كليـــــــة العلوم
السملالية - مراكش
FACULTÉ DES SCIENCES
SEMLALIA - MARRAKECH

Mémoire de projet de fin d'étude

En vue de l'obtention du diplôme de Licence Fondamentale
Spécialité : Sciences Mathématiques et informatique

APPRENTISSAGE NON SUPERVISE POUR L'IDENTIFICATION DES INFLUENCEURS DANS UN RESEAUX SOCIAL

Soutenu le 04/07/2022

Réalisé par :

KHALIDI WALID

SALFI YASSINE

Encadré par :

Pr. QAFFOU ISSAM

Année Universitaire : 2021/2022

Dédicace

À tous les parents du monde qui ont goûté à l'amertume de la vie pour enseigner à leurs enfants le sens de la persévérance,
Aux plus chères créatures de Dieu, nos parents qui nous ont aimés sincèrement et qui ont tant sacrifié pour répondre à nos demandes et à nos ordres et établir des valeurs morales en nous.
À nos frères qui nous ont enseigné avant l'école, qui à leur tour ont sacrifié leur temps et leur argent pour nous fournir tout ce dont nous avons besoin et nous ont grandement encouragés dans notre cheminement éducatif. Dans notre vie en général.
Aux enseignants du monde qui luttent pour assurer une bonne éducation pour les générations futures, en particulier les professeurs qui nous ont enseignés, nous ont dirigés, nous ont beaucoup appris à établir l'éthique et les méthodes de réussite, et se sont tenus à nos côtés dans les moments les plus forts pour nous fournir une bonne éducation.
À nos amis qui se sont tenus à nos côtés dans les moments de joie et les moments d'amertume, et nous ont soutenus par leurs pensées et leur temps.
À tous ceux qui s'accrochent à la vie et se battent pour ses fils et amis.

Remerciements

La réalisation de ce mémoire a été possible grâce au concours de plusieurs personnes à qui nous voudrions témoigner toute notre gratitude.
Nous voudrions tout d'abord adresser toutes nos reconnaissances à l'encadrant de ce mémoire, Mr QAFFOU Issam, pour sa patience, sa disponibilité et surtout ses judicieux conseils, qui ont contribué à alimenter nos réflexions.
Nous désirons aussi remercier les professeurs de la faculté des sciences Semlalia, qui nous ont fourni les outils nécessaires à la réussite de nos études universitaires.
Nous voudrions exprimer nos reconnaissances envers les amis et collègues qui nous ont apporté leur soutien moral et intellectuel tout au long de notre démarche.
Nous adressons nos vifs remerciements aux membres des jurys pour avoir bien voulu examiner et juger ce travail.

Résumé

Identifier les nœuds influents est un sujet essentiel dans de nombreuses applications, notamment accélérer la propagation des faits, contrôler les rumeurs et les maladies. Plusieurs méthodes ont été proposées pour identifier les nœuds d'influence dans un réseau complexe, allant de la centralité des nœuds aux processus basés sur la diffusion.

Dans ce projet, une nouvelle méthode d'évaluation de l'importance de nœud dans les réseaux complexes basée sur une technique de performance d'ordre par similarité à la solution idéale (TOPSIS) approche est proposée. TOPSIS est d'abord appliqué pour identifier les nœuds influents dans un réseau complexe dans ce problème ouvert. De diverses façons, on considère plusieurs mesures de centralité comme le multi-attribut de réseaux complexes dans l'application TOPSIS.

TOPSIS est employé pour agréger le multi-attribut pour obtenir une évaluation de la taille de nœud de chaque nœud. Elle ne se limite pas à une seule mesure de centralité, mais envisage différentes mesures de centralité car chaque mesure de centralité a ses propres inconvénients et limites. Ensuite, nous proposons d'utiliser l'algorithme K-Means pour trouver des clusters dont les centroïdes sont les influents Top-K trouvés par la méthode TOPSIS.

En fin de compte, nous utilisons le modèle SI (Susceptible-Infected) pour évaluer le rendement.

Mots-clés : Réseaux sociaux, théorie des graphes, mesures de centralité, apprentissage automatique, détection des nœuds influents.

Table des matières

Liste des figures

Liste des tableaux

Introduction générale

La détection des nœuds influents dans les réseaux sociaux a suscité beaucoup d'intérêt dans la communauté des chercheurs. Dernièrement, de nombreuses techniques ont été proposées pour trouver des nœuds influents dans des réseaux complexes. La connaissance de la capacité de propagation du nœud montre de nouvelles perspectives d'application, comme le contrôle de la propagation des messages et des rumeurs dans les réseaux sociaux, le classement de la réputation des scientifiques, etc.

Ainsi, la détection des nœuds influents permet beaucoup d'avantages dans l'analyse et la prise de décision dans un réseau complexe. Pour cet objectif, l'algorithme proposé dans le présent rapport essaie de trouver les nœuds les plus influents dans un réseau donné en se basant sur des critères multiples (mesures de centralité). Dans les travaux existants, plusieurs mesures de centralité ont été proposées pour identifier les nœuds influents (Betweeness, degré, closeness, et eigenvector centrality). Elles ont tous mis l'accent sur une mesure de centralité et elles ont certaines limites. Dans ce rapport, nous allons démontrer que les mesures de centralité ont des performances différentes pour trouver les nœuds influents. Si une seule mesure de centralité est adoptée, l'identification des nœuds influents pourrait être différente si on utilise une autre mesure de centralité. Ainsi, pour faire face à cette inefficacité dans la recherche des nœuds les plus influents dans les réseaux sociaux, l'une des techniques de prise de décisions multi-attributs est utilisée dans notre travail.

Le rapport suivant est étalé sur quatre chapitres. Le premier chapitre présente le contexte général avec des notions de bases. Le chapitre 2 discute les méthodes appliquées et la démarche suivie. Le chapitre 3 présente les outils et les technologies utilisées. Le chapitre 4 détaille la mise en œuvre et réalisation de notre projet. Vers la fin, le rapport sera conclu en ouvrant quelques perspectives.

Chapitre 1 : Contexte général

Introduction

Dans ce chapitre, nous allons présenter le contexte général de notre projet on se base sur les questions suivant : c'est quoi un réseau complexe ? Et en particulier un réseau social ? Quels sont les types de ces réseaux ? Quelle sont leurs caractéristiques ? Pour répondre a c'est questions nous allons donner des exemples de ces réseaux. Dans la deuxième section nous allons introduire la notion de la théorie des graphes et comment les utilise pour modéliser les réseaux complexes. Enfin dans la dernière section nous parlons des nœuds influents et pourquoi on a besoin de les identifier dans un réseau complexe.

1.1. Réseaux complexes

Les réseaux complexe sont présents dans plusieurs domaines divers comme : biologie, sociologie, psychologie, informatique...Ils recouvrent la majorité des réseaux comme le réseau internet, les réseaux humains, les réseaux de protéine, etc. c'est réseaux peuvent être classé on quatre catégories :

- Les réseaux sociaux
- Les réseaux d'informations
- Les réseaux technologiques
- Les réseaux biologiques

a. Réseau social

On peut définir ce type de réseau comme un ensemble de personnes en contact ou en interaction entre eux et qui sont présenter sous la forme d'un schéma (Figure1.1.1).

Figure 1.1.1 Exemple d'un réseau

b. Réseau informatique

L'exemple Classique de cette catégorie est le réseau internet qui relie plusieurs page web entre eux à l'aide des Canales on peut représenter les pages comme des nœuds et les Canales comme des arêtes (Figure 1.1.2).

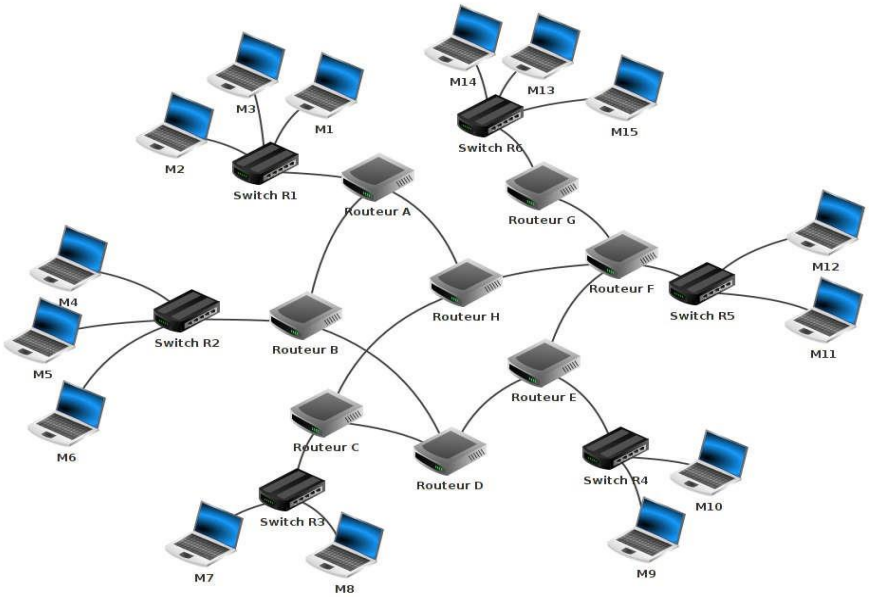

Figure 1.1.2 Exemple d'un réseau informatique

c. Un réseau technologie

Ce réseau créé par l'homme à la raison de distribuer des services ou de l'énergie à l'aide des lignes et des Canales par exemple les réseaux électriques, les réseaux aériens, etc. (Figure 1.1.3)

Figure 1.1.2 Exemple d'un réseau technologie

d. Un réseau biologique

C'est un réseau lié à l'être vivant. Un exemple de réseau biologique est le réseau nerveux. (Fig1.1.4)

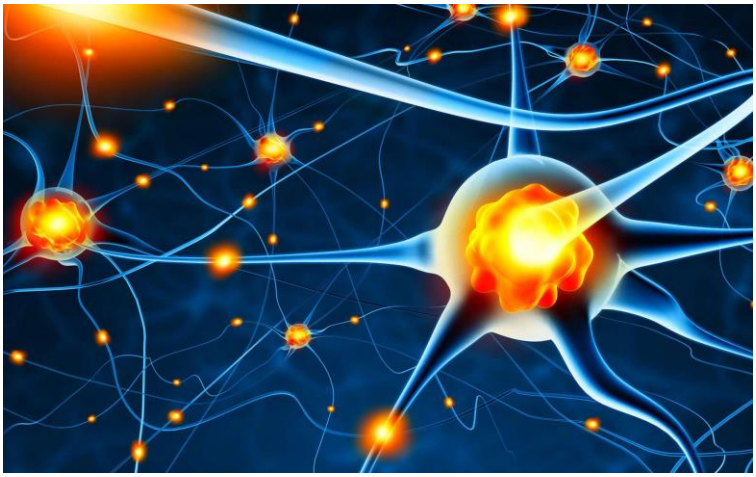

Figure 1.1.4 Exemple d'un réseau biologique

1.2. Modélisation par théorie des graphes

La théorie des graphes est un outil adéquat pour représente les réseaux. Différentes études sont utilisées ces outils pour la modélisation des réseaux complexe. Dans notre cas, la théorie des graphes sera représentée, puis on étudiera les caractéristiques des propriétés structurelles d'un graphe pour arriver au problème de la dynamique, et aussi les limites auxquelles on se confronte à l'heure actuelle. Pour y parvenir, nous utilisons la théorie des graphes.

Par définition, un graphe est un outil permettant de représenter un ensemble d'éléments pouvant être reliés les uns aux autres. Ces éléments sont appelés des *nœuds* et les liens qui les relient des *arcs*.

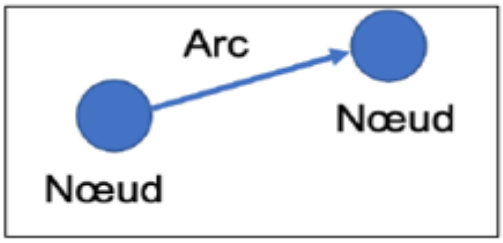

Figure 1.2.1 : Elément de base d'un graphe

Dans un graphe d'un réseau social, les nœuds représentent des individus et les arcs indiquent des liens entre ces individus. Chaque graphe est constitué des éléments suivant :

a. *Complexité du graphe*

La *complexité* du graphe dépend de la possibilité de trouver plusieurs arcs entre deux nœuds *voisins*, c'est-à-dire deux nœuds directement reliés par un arc. Un graphe est *simple* lorsque deux nœuds sont reliés par au maximum un seul arc et qu'il n'y a pas de boucle. Un graphe est un *multi graphe* lorsque deux nœuds peuvent être reliés par plus d'un arc ou qu'il y a une boucle. Cette distinction permet d'étudier l'importance de la multiplicité des interactions sociales dans l'action des influenceurs.
Un exemple de *graphe simple* est un graphe d'amitié issu de Facebook : la relation d'amitié entre deux individus est unique, alors, il n'est pas possible de trouver plus d'un arc entre deux nœuds. Au contraire, un graphe de retweet issu de Twitter peut être un multi graphe dans le cas où un utilisateur a retweeté plusieurs tweets d'un autre utilisateur. Nous représentons en figure 3 un exemple pour chacun des deux types de graphe.

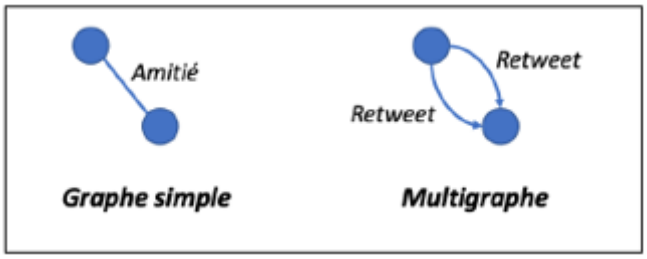

Figure 1.2.2 : Illustration des concepts de graphe simple et de multi graphe

b. *Orientation du graphe*

Un graphe est dit orienté si chacun de ses arcs est orienté, c'est-à-dire avec un nœud source et un nœud cible (cf. Figure 4). Dans le cadre de l'étude des influenceurs, cette propriété du graphe est déterminante puisqu'elle permet en particulier d'identifier la source d'une action d'influence. Nous analyserons donc essentiellement des graphes orientés.

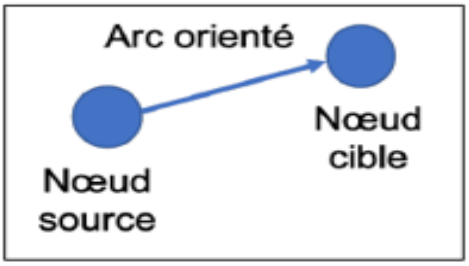

Figure 1.2.3 : Illustration du graphe oriente

c. *Degré d'un nœud*

Le *degré d'un nœud* correspond au nombre d'arcs qui lui sont incidents. Dans un graphe orienté, le *degré entrant* ne tient compte que des arcs orientés vers le nœud considéré tandis que le *degré sortant* ne tient comptent que de ceux orientés vers d'autres nœuds. Les mesures de centralité se basent principalement sur cette propriété parce qu'elle rend compte de l'intégration d'un nœud dans un graphe.

1.3. Détection des nœuds influents

Les influenceurs ont la capacité d'avoir un impact sur d'autres individus lorsqu'ils interagissent avec eux. Détecter les influenceurs permet d'identifier les individus à cibler pour toucher largement un réseau. Il est possible d'analyser les interactions dans un média social du point de vue de leur structure ou de leur contenu. Dans notre travail, nous présentons d'abord une évaluation de différentes mesures de centralité sur la structure d'interactions extraites de FACEBOOK puis nous analysons l'impact de la taille du graphe de suivi sur la performance de mesures de centralité.

Conclusion

L'importance des nœuds est une mesure de base pour caractériser la structure et la dynamique des réseaux complexes. Par conséquent, l'identification des nœuds influents a été une question ouverte et une tâche de recherche critique dans les réseaux complexes. Diverses mesures de centralité ont été proposées au fil des ans pour classer les nœuds d'un graphique en fonction de leur importance topologique.

Chapitre 2 : Méthodes appliquées

Introduction

Pour implémenter et évaluer les réseaux complexe de notre projet pour détecter les Influenceurs, nous avons choisi d'utiliser la méthode de MCDM, et l'algorithme d'apprentissage non superviser.et aussi nous avons utilisés le model SI pour évaluer les résultats.

2.1. Méthodes MCDM

Il existe des techniques de prise de décision multicritère (Multi-Criterion Decision Making) pour résoudre des problèmes impliquant la sélection d'alternatives basées sur des critères contradictoires. La technique d'ordre de préférence par similarité avec la solution idéale (TOPSIS) est une technique MCDM populaire ayant une gamme variée de domaines d'application en raison de sa facilité de mise en œuvre ainsi que de sa nature intuitive.

TOPSIS (Technique for ordre préférence by similarity to ideal solution) méthode est appelée la solution idéale. Il s'agit d'une méthode efficace de prise de décision à attributs multiples. Cette méthode consiste à construire les solutions idéales et les solutions moins idéales aux problèmes des attributs multiples et utilise les deux repères d'être proche des solutions idéales et d'être loin des solutions moins idéales comme critères d'évaluation des projets réalisables. "Solution idéale" et " solution moins idéale" sont les deux concepts de base de la méthode TOPSIS.

La solution dite idéale (notée x+) si la solution hypnotiquement optimale, toute sa valeur d'attribut atteint la meilleure valeur de chaque solution alternative ; mais la solution moins idéale (notée x-) est la pire solution dans l'hypothèse.

La règle de classement des solutions est de comparer chaque solution alternative avec x+ et x-. Si l'un des solutions est proche de x+ et loin de x- au même temps, alors il est la meilleure solution parmi les solutions alternatives. Les étapes de l'application de la méthode TOPSIS sont les suivants :

- **Première étape :**

Construire une matrice de décision normalisée A. Pour les questions d'évaluation complète avec n unités d'évaluation et m indices d'évaluation, sa matrice de décision A est :

$$A = \begin{array}{c} \\ x_1 \\ x_2 \\ \cdots \\ x_n \end{array} \begin{array}{c} \begin{array}{cccc} f_1 & f_2 & \cdots & f_m \end{array} \\ \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \cdots & \cdots & \cdots & \cdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{bmatrix} \end{array}$$

Dans la formule, aij=fj(xi), qui montre l'indice d'évaluation j'ème de la i'ème unité d'évaluation (projet alternatif). i=1,…, n ; j=1,2,…, m. Standardiser la matrice A comme matrice R :

$$R = \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1m} \\ r_{21} & r_{22} & \cdots & r_{2m} \\ \cdots & \cdots & \cdots & \cdots \\ r_{n1} & r_{n2} & \cdots & r_{nm} \end{bmatrix}$$

Avec $r_{ij} = \frac{x_{ij}}{\sqrt{\sum_{j=1}^{m} x_{ij}^2}}$, $i=1,\ldots,n;\ j=1,\ldots,m.$

- **Deuxième étape :**

Construire la matrice de décision pondérée et normalisée V, le vecteur de poids :

W= (w1,w2,…,wn).

$$V = \begin{bmatrix} v_{11} & v_{12} & \cdots & v_{1m} \\ v_{21} & v_{22} & \cdots & v_{2m} \\ \cdots & \cdots & \cdots & \cdots \\ v_{n1} & v_{n2} & \cdots & v_{nm} \end{bmatrix}$$

Avec : vij=wj.rij, i=1,2,…, *n*, *j*=1,2,…, *m*.

- **Troisième étape :**

Déterminer la solution idéale x+ et moins solution idéale x- :

x+ = {max vij |jϵJ), (min vij | jϵJ ′) |*i*=1, 2,…, *n*}={x1+, x2+,…, xm+}
x- = {min vij |jϵJ), (max vij | jϵJ ′) |*i*=1, 2,…, *n*}={x1-, x2-,…, xm-}

- **Quatrième étape :**

Calculer la distance de chaque solution par à la solution idéale x+ est :$S_i^+ = \sqrt{\sum(v_{ij} - x_j^+)^2}$

La distance de chaque projet à la solution moins idéale x- est : $S_i^- = \sqrt{\sum(v_{ij} - x_j^-)^2}$

- **Cinquième étape :**

Calculer l'indice de proximité relatif de chaque projet à la solution idéale ci : $C_i = \frac{S_i^-}{S_j^+ + S_i^-}$

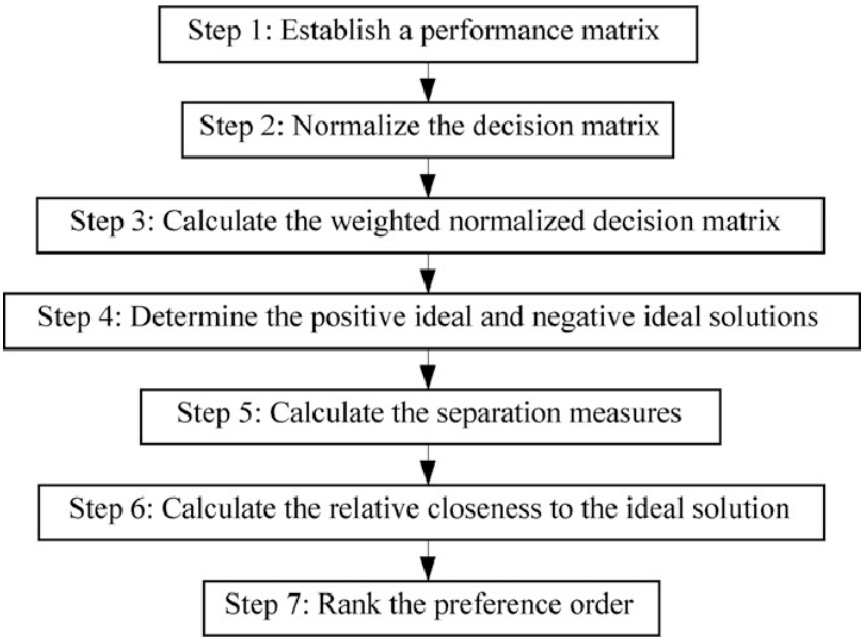

Figure 2.1 : les différentes étapes de la méthodes TOPSIS

2.2. Apprentissage non supervisé

L'apprentissage non supervisé est celui où l'algorithme doit opérer à partir d'exemples non annotés. En effet, dans ce cas de figure, l'apprentissage par la machine se fait de manière entièrement indépendante. Des données sont alors renseignées à la machine sans qu'on lui fournisse des exemples de résultats.

Dans ce cadre, l'ensemble des données collectées est traité comme des variables aléatoires. En effet et contrairement à l'apprentissage automatique qui se doit de trouver un modèle à partir de données étiquetées : f(X) Y, il utilise seulement des données non étiquetées : il n'y a pas de variable Y à prédire.

L'utilisation de l'apprentissage non supervisé peut être réunie en problèmes de clustering et d'association.

Un problème de clustering est un problème pour lequel on attend de la machine qu'elle **rassemble sous forme de groupe (mise en cluster)** des objets présents dans des groupes de données, et ce de la manière la plus juste et efficace possible. Cette technique, bien que parfois difficile à comprendre par l'homme, est très utilisée dans le domaine du marketing pour placer dans des groupes les différents clients par exemple.

Un exemple d'algorithme très souvent utilisé dans le clustering est le **k-moyennes** qu'on appelle **k-means**.
L'algorithme de k-means est l'une des techniques de regroupement les plus populaires dans les tâches d'apprentissage non supervisé.
Compte tenu d'un ensemble de nœuds ou de bus, cet algorithme a été utilisé efficacement pour diviser un réseau en k clusters. Ceci est basé sur le placement optimal du centroïde pour le cluster respectif dans un réseau.

Dans cet algorithme, le réseau est initialement divisé en k cluster, chaque cluster étant définie par un bus de référence (centroïde). Les autobus restants sont ensuite répartis et affectés de façon appropriée aux clusters en fonction de la proximité de chaque autobus par rapport aux autobus de référence. Ensuite, des ajustements de clusters sont effectués avec le calcul de nouveaux centroïdes. Ces centroïdes servent de nouveaux points de référence pour le prochain partitionnement de tous les bus. Ces ajustements produisent naturellement un minimum d'erreurs qui correspond à la « configuration de Voronoi », ce qui donne des emplacements de référence au centroïde des clusters. La mesure d'erreur ou la fonction potentielle est la somme de tous les écarts et est donnée comme indiqué :

$$\phi = \min \sum_{j=1}^{k} \sum_{i=1}^{n_j} |x_{ij} - \mu_j|^2,$$

Le processus devient itératif afin que les clusters atteignent un minimum local qui dépend de la sélection initiale des bus de référence. L'algorithme k-means continue à ajuster les centroïdes après chaque partition, ce qui le rend plus dynamique aux changements.

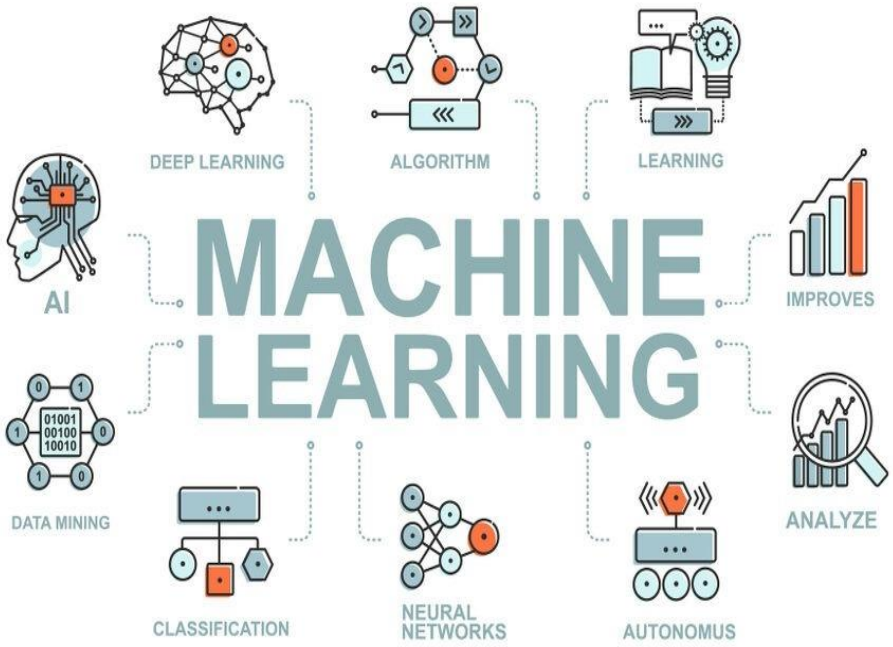

Figure 2.2 : Différente méthodes de Machine Learning

Algorithme k-means

Entrée :

K le nombre de cluster à former

Le training set (matrice de données)

Début :

Choisir aléatoirement K point (une ligne de la matrice de données). Ces points sont les centres des clusters (nomme centroïde).

Répéter

Affectation de chaque point (Une ligne de la matrice de données) au groupe dont il est le plus proche à son centre

Recalculer le centre de chaque cluster et modifier le centroide

Jusqu'à convergence ou stabilisation de l'inertie totale de la population

Fin de l'algorithme

2.3. Modèle SI

Tout d'abord, nous en déduisons une équation qui représenterait comment une infection se propagerait théoriquement dans un réseau. Pour générer une équation théorique pour un réseau SI temporel, nous examinons la relation entre S et I, qui est S + I = 1. Ensuite, nous calculons la probabilité pour chaque combinaison de bords, qui pourrait être SI, SS, ou II. Après les probabilités de chaque combinaison possible sont trouvées, la valeur attendue pour le nombre de nœuds infectés à l'étape suivante est déterminée. Puisque les combinaisons de nœuds SS et II n'augmenteront pas dans la fraction infectée, les combinaisons sont classées comme 0, parce que 0 nœuds deviennent infectés. La dernière combinaison, SI, entraînera l'infection d'un nœud supplémentaire, de sorte que la proportion de nœuds infectés augmente avec 1n. Le taux de changement du nombre de nœuds infectés est alors écrit comme suit :

$$\frac{dI}{dt} = \begin{cases} \frac{0}{n} & \text{with probability } (1-I)^2 + I^2 \\ \frac{1}{n} & \text{with probability } 2I(1-I) \end{cases}$$

A partir de là, nous écrivons la fonction de valeur attendue pour le nombre de nœuds infectés, qui vient de l'idée que la valeur attendue est équivalente à la somme des possibles

$$\frac{dI}{dt} = \frac{2}{n} I(1-I).$$

Résultats multipliés par leurs probabilités. Le changement attendu des nœuds infectés simplifie : Nous passons ensuite par une dérivation similaire pour activer deux arêtes même temps, puis trois, et concevons une théorie générale des champs moyens basée sur le nombre de arêtes activées en même temps (w) :

$$\frac{dI}{dt} = \frac{2w}{n} I(1-I).$$

Dans cette dérivation, les bords qui partagent un nœud commun sont ignorés. Nous simulons ensuite la propagation de l'infection sur un réseau temporel sous les mêmes hypothèses que le modèle et nous le comparons à notre équation théorique.

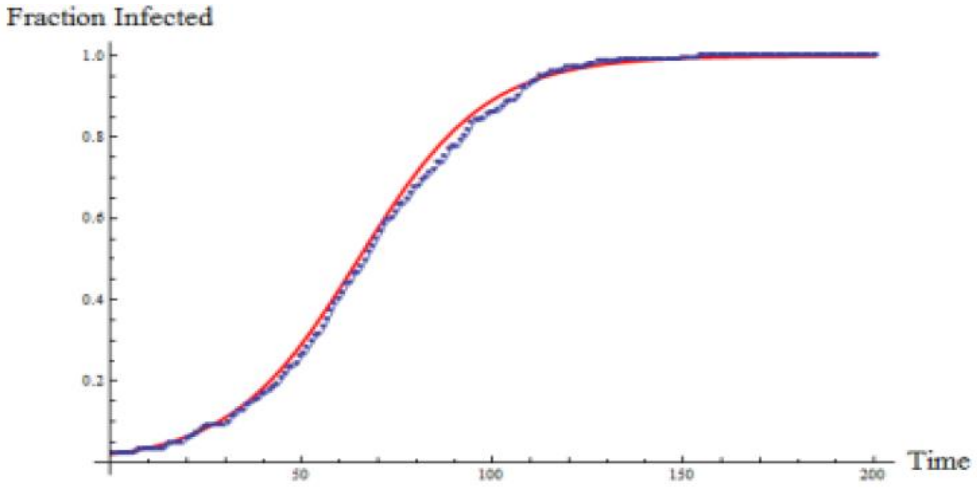

Figure 3.1 : La courbe de champ moyenne théorique de l'équation 2 est en rouge et la ligne pointillée bleue est la courbe médiane de différents

Comme le montre la Figure 3.1, la médiane des courbes expérimentales et la courbe théorique sont assez similaires. Nous utilisons la médiane plutôt que la moyenne parce qu'il y a des valeurs aberrantes qui fausseraient la moyenne, soit plus élevée ou plus faible que ce qui est prévu. Cela montre que le réseau temporel correspond étroitement à notre prédiction théorique de la façon dont la maladie se propagerait à travers le réseau temporel.
Après avoir examiné la théorie du champ moyen temporel, nous remarquons que certaines combinaisons de bords pourraient partager des nœuds. Cela pourrait alors permettre aux voisins des voisins d'un nœud infecté de devenir infecté en une seule étape de temps, si l'infection a été autorisée à se propager d'un nœud infecté à un nœud sensible et à travers une autre connexion à un nœud sensible différent. Ainsi, nous examinerons plusieurs cycles d'infection pour un réseau.

2.4. Démarche d'application

Dans ce projet on va suivre des étapes l'un après l'autre pour attendre notre but. Premièrement nous avons appliqué topsis, avec les calculs des mesures de centralité de centralités sur les data sets qu'on a, après nous avons classe les résultats pour citer les tops 10, deuxième étape c'est l'application de l'algorithme k_means, dans cette étape nous avons initialise les conoïdes de l'algorithme k_means par les tops 10 qui nous avons obtenu d'après l'application de topsis, Troisième étapes nous avons changé la façon utilisable de l'algorithme k_means pour calculer la distance entre les nœuds par une autre façon, nous avons utilisé les score d'un nœud pour calculer la distance, à l'aide des mesures de centralités de chaque nœud nous avons réalisé les relations suivant pour définir nos scores :
Pour les petits réseaux (ex : football, Zachary…) : SN= (BC+DC+CC+EC)/4
Pour les grandes réseaux (ex : Facebook) : SN= (BC+DC+CC)/3

Quatrième étapes nous avons comparé à l'aide de SI model les résultats qui nous avons obtenu dans chaque étape, pour fine notre projet par une conclusion générale qui contentaient les résultats globaux de notre projet.

Conclusion

L'identification des influenceurs dans un réseau complexe peut se faire par diffèrent méthode, parmi les méthodes nous avons choisi la méthode de topsis pour avoir une liste des tops 10 de chaque réseau, ils ont un rôle très important dans l'implémentation de l'algorithme de k-means qui se classe dans les algorithmes de l'apprentissage non superviser.

Chapitre 3 : Outils et technologies utilisés

Introduction

Pendant la réalisation de ce projet nous avons utilisé des outils informatiques qui nous ont aidés pour implémenter de manières efficace et simple les différentes méthodes et algorithme. Nous avons utilisé le langage python pour défier notre code source qui contient des bibliothèques spécifier pour l'intelligence artificielle, et à l'aide des distributeurs de langage de programmation ANACONDA et Google colab.

3.1. Ressources logicielles

a. Le langage python

Pour défier notre code source, qui est un langage de programmation interprété multi-paradigme. Il favorise la programmation impérative structurée, et orientée objet. Il est doté d'un typage dynamique fort, d'une gestion automatique de la mémoire par ramasse-miettes et d'un système de gestion d'exceptions ; il est ainsi similaire à Perl, Ruby, Schème, Small talk et Tcl.

Figure 3.1.a : Logo python

b. Anaconda

C'est une distribution libre et open source des langages de programmation Python et R appliqué au développement d'applications dédiées à la science des données et à l'apprentissage automatique, qui vise à simplifier la gestion des paquets et de déploiement. Les versions de paquetages sont gérées par le système de gestion de paquets *conda*.

Figure 3.1.b : Logo ANACONDA

c. Google colab

Cet outil nous permet de tirer pleinement parti des bibliothèques populaires Python pour analyser et visualiser des données.

Figure 3.1.c : Logo Google Colab

3.2. Package utilisées

a. NetworkX

C'est un package de langage de python pour la création, manipulation, et l'étude de la structure, Dynamics, et fonction des réseaux complexe.il est utilisé pour étudier les réseaux complexe qui est représenter sous la forme de graphe avec nœud et bords. L'utilisation de network permet de charger et de stocker des réseaux complexes. Nous pouvons générer de nombreux types de réseaux aléatoires et classiques, analyser la structure de réseau, construire des modelés de réseau, concevoir de nouveaux algorithmes de réseau et dessiner des réseaux.

Figure 3.2.a : Logo de bibliothèque Networks

b. Mathplotlib

C'est une bibliothèque de traçage pour créer des visualisations statiques, animées et interactives en Python. Matplotlib peut être utilisé dans les scripts Python, le Shell Python et Python, les serveurs d'applications web, et divers outils d'interface utilisateur graphique comme Tkinter, awxPython, etc.

Figure 3.2.b : Logo de bibliothèque Mathplotlib

c. Pandas

C'est une bibliothèque open-source qui est construite au-dessus de la bibliothèque NumPy. C'est un paquet Python qui offre diverses structures de données et opérations pour manipuler des données numériques et des séries chronologiques. Il est principalement populaire pour importer et analyser des données beaucoup plus faciles. Pandas est rapide et il a de hautes performances et de productivité pour les utilisateurs.

Figure 3.2.c : Logo de bibliothèque Pandas

d. Numpy

C'est une bibliothèque Python utilisée pour travailler avec des tableaux.
Il a également des fonctions pour travailler dans le domaine de l'algèbre linéaire, transformée de Fourier, et les matrices.
NumPy a été créé en 2005 par Travis Oliphant. C'est un projet open source et vous pouvez l'utiliser librement.

Figure 3.2.e : Logo de bibliothèque Numpy

e. NDLIB

C'est un package Python qui permet de décrire, simuler et étudier les processus de diffusion sur des réseaux complexes.

Figure 3.2.f : Logo de bibliothèque NDlib

f. scikit-learn

C'est une bibliothèque Python open-source qui implémente une gamme d'algorithmes d'apprentissage automatique, de prétraitement, de validation croisée et de visualisation utilisant une interface unifiée.

Figure 3.9.g : Logo de bibliothèque Sckit.lear

Chapitre 4 : Mise en œuvre et réalisation

Introduction

La mise en œuvre et la réalisation des différentes taches de ce projet a passé par plusieurs étapes lors de l'implémentation avec les outils déjà définis dans le chapitre 3. Dans la première partie nous allons présenter les réseaux utilisés dans ce travail, par la suite nous allons appliquer la méthode TOPSIS mais d'abord nous définissons la notion des mesures de centralités. Dans la troisième section le regroupement des nœuds sera fait par l'algorithme de k-means. La derniers nous allons définir la notion de SI model qui va nous aide pour discuter les résultats.

4.1. Réseaux utilisés

Nous avons utilisé différents réseaux pour avoir une vision sur l'effet de chaque méthode. D'abord on a commencé par l'utilisation du réseau Facebook Ego qui contient 4038 nœuds et 88234 arêtes ensuite nous avons passé à un réseau moins complexe qui s'appelle Email qui contient 1133 nœuds et 5451 arêtes. Par la suite nous avons appliqué les méthodes sur des petits réseaux ; le premier est Football qui contient 115 nœuds et 613 arêtes, le deuxième est Zachary qui contient 34 nœuds et 78 arêtes et le dernier Dolphins qui contient 62 nœuds et 192 arêtes Tous ces réseaux sont stocké dans des fichiers sous la forme de deux colonnes et des lignes. Ces derniers contiennent les numéros des nœuds.

Ces réseaux sont représentés graphiquement dans les figures suivantes.

- **Pour le réseau Facebook ego :**

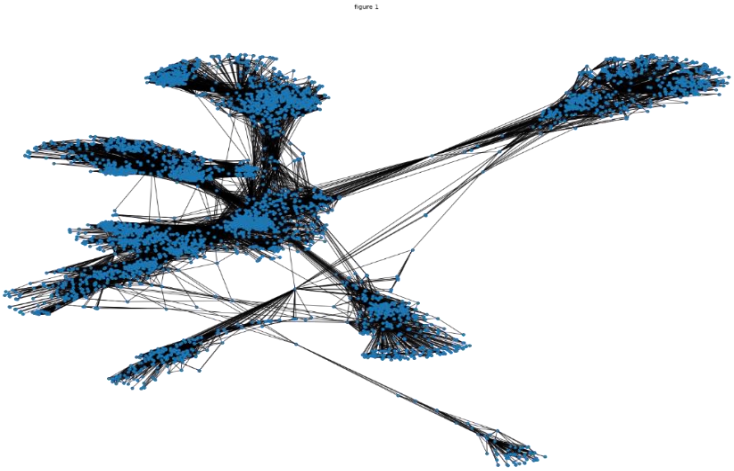

Figure 4.1.1 : Graphe de réseau Facebook ego

- **Pour le réseau Email :**

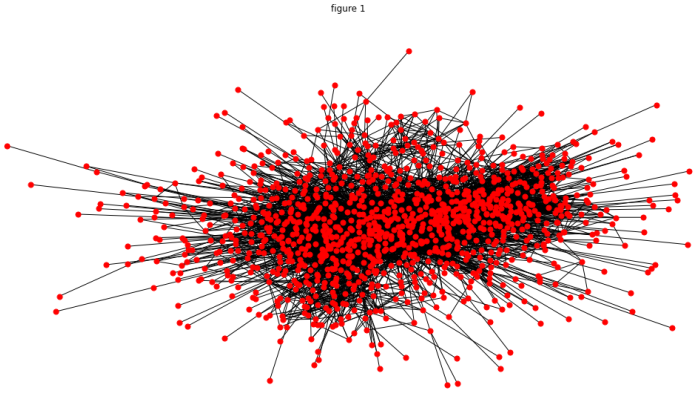

Figure 4.1.2 : Graphe de réseau Email

- **Pour le réseau Football :**

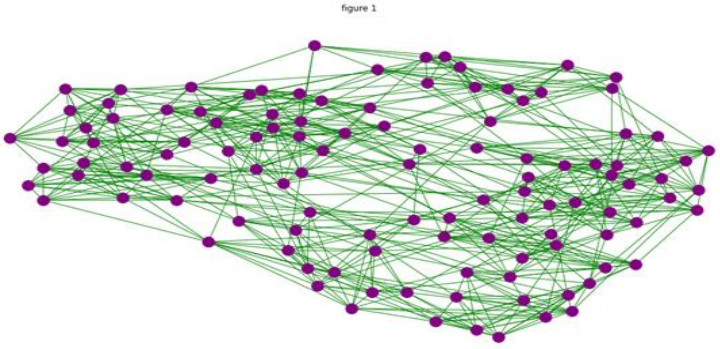

Figure 4.1.3 : Graphe de réseau Football

- **Pour le réseau Zachary :**

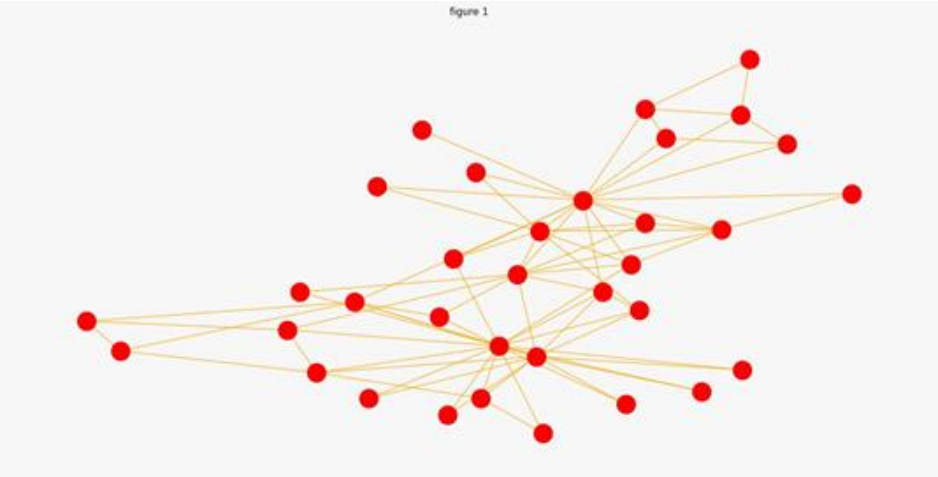

Figure 4.1.4 : Graphe de réseau Zachary

- **Pour le réseau Dolphins :**

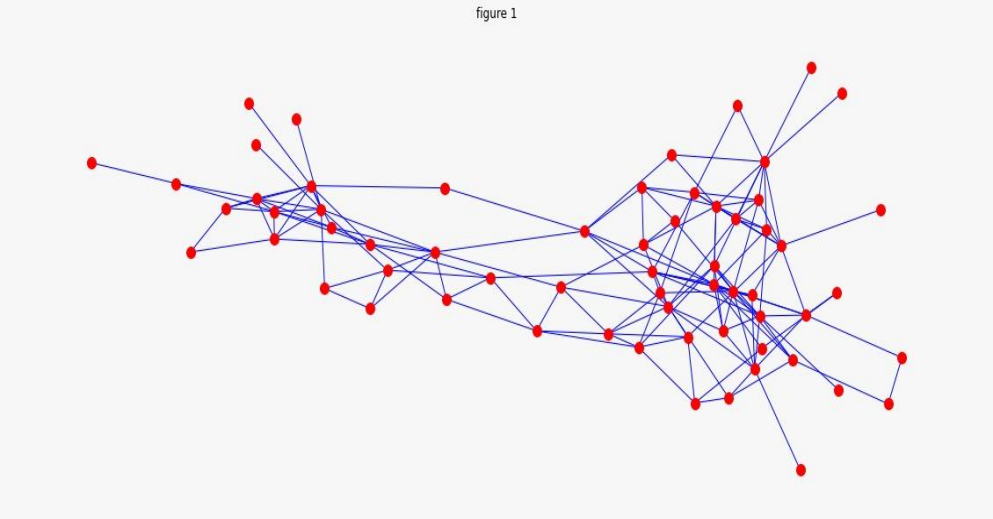

Figure 4.1.5 : Graphe de réseau Dolphins

4.2. Application de TOPSIS

L'objectif de notre travail est la détection des nœuds influents dans un réseau social. Comme première étape, on va appliquer la célèbre méthode TOPSIS qui permet de faire le classement des nœuds dans un réseau suivant leur influence. Pour ce faire, les calculés seront basés sur les mesures de centralités.
De nombreuses mesures de centralité ont été proposées pour classer les nœuds dans les réseaux. Un simple est la centralité de degré, à savoir, un nœud avec un degré plus grand est susceptible d'avoir une influence plus élevée (par exemple, comme un nœud initialement infecté, il est

prévu de se propager plus rapidement et plus largement) qu'un nœud avec un degré plus petit. Cependant, dans certains cas, ne parvient pas à identifier les nœuds influents car elle ne prend en compte que des informations très limitées. Un autre groupe de méthodes tenant compte de l'information globale donne de meilleurs résultats de classement, comme betweenes centrality et closeness centrality de classement par voie.

- **Degree centrality**

Le degré de centralité est l'un des plus faciles à calculer. La centralité de degré d'un nœud est simplement son degré — le nombre de bords qu'il a. Plus le degré est élevé, plus le nœud est central. Cela peut être une mesure efficace car de nombreux nœuds avec des degrés élevés ont aussi une centralité élevée à travers d'autres mesures.
Dans certains cas, cette méthode n'arrive pas à identifier les nœuds influents a cause de nombre limitées des informations qui prend en compte. Par exemple dans la figure 1 le nœud 1 a le plus grand dégrée par apport ou autre nœuds, si en suppose qu'une maladie infecte le nœud 1 la propagation ne peut être rapide et large. En revanche, le nœud 23 peut avoir une influence plus élevée, vu qu'il a un dégrée plus faible, donc on est besoin d'autre méthodes tenant en compte des informations plus détaille sur les nœuds.

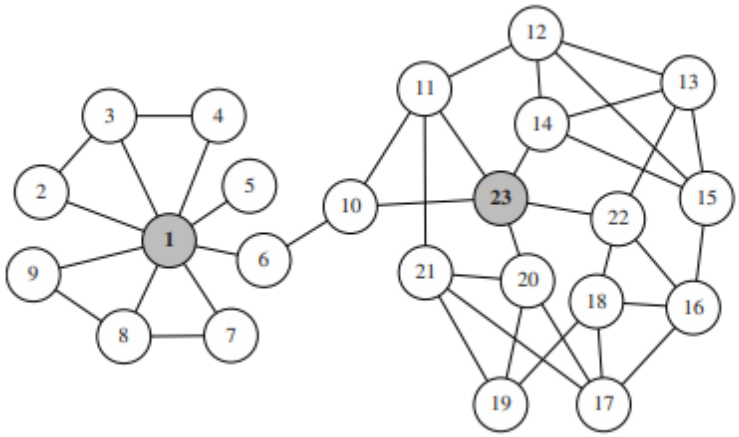

Figure 4.2.1 : Exemple d'un réseau contient 23 nœuds et 40 arêtes

- **Betweenness centrality**

Betweenness est une mesure de centralité d'un nœud dans un réseau, généralement définie comme la fraction de chemins les plus courts entre les paires de nœuds qui passent par le nœud d'intérêt.
Entre-temps, il s'agit, dans un certain sens, d'une mesure de l'influence d'un nœud sur l'information diffusée par le réseau ou sur la charge attendue d'un nœud dans un réseau de transport.
Pour un réseau G = (V, E) avec n = |V| nœuds et m = |E| frontières, la centralité entre nœuds v, indiquée par CB(v) est :

$$C_B(v) = \sum_{s \neq v \neq t \in V} \frac{\sigma_{st}(v)}{\sigma_{st}},$$

Où σ_{st} est le nombre de chemins les plus courts entre les nœuds s et t, et $\sigma_{st}(v)$ indique le nombre de chemins les plus courts entre s et t qui passent par le nœud.

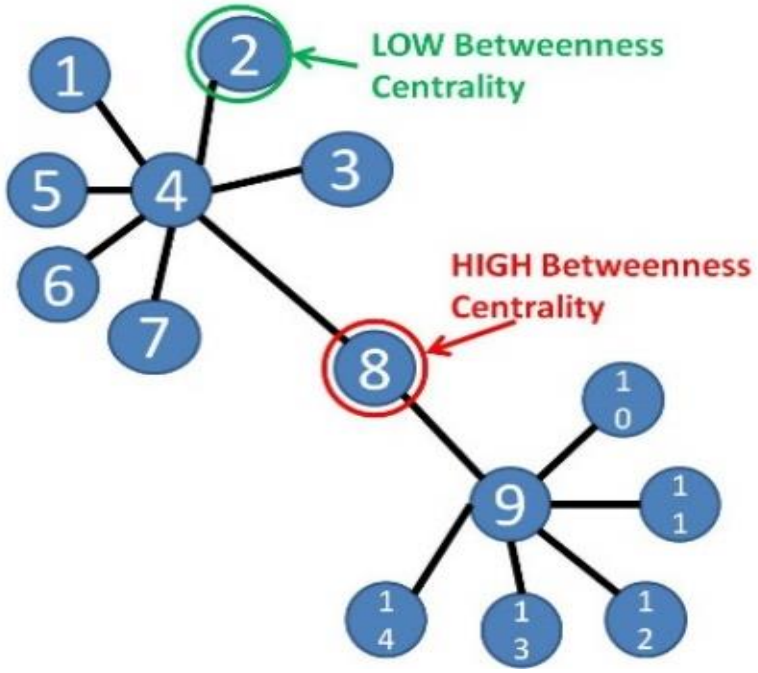

Figure 4.2.2 : Exemple explicative de la mesure de centralité BC

- **Closeness centrality**

Closeness du nœud v est définie comme la réciproque de la somme des distances géodésiques à tous les autres nœuds de V :

$$C_C(v) = \frac{1}{\sum_{t \in V \setminus v} d_G(v,t)},$$

Où $d_G(v,t)$ est la distance géodésique entre v et t. Closeness peut être considérée comme une mesure de la durée pendant laquelle l'information sera diffusée d'un nœud donné à d'autres nœuds accessibles dans le réseau.

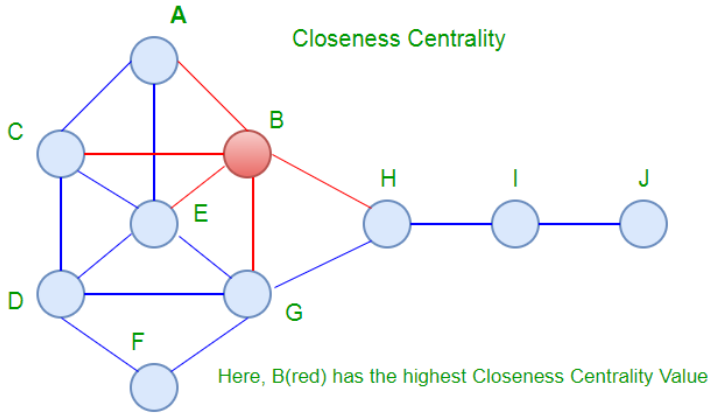

Figure 4.2.3 : Exemple explicative de la mesure de centralité CC

- **Eigenvector centrality**

Eigenvector centralité est une mesure de l'influence d'un nœud dans un réseau. Il attribue des scores relatifs à tous les nœuds du réseau en fonction du concept que les connexions aux nœuds à haut score contribuent davantage au score du nœud en question que les connexions égales aux nœuds à faible score.

Qu'A soit une matrice de similarité n x n.

Eigenvector (vecteur propre) xi du nœud i est définie comme l'Ième entrée dans le vecteur propre normalisé appartenant À la plus grande valeur propre d'A. λ est la plus grande valeur propre de A et n est le nombre de sommets :

$$Ax = \lambda x, \qquad x_i = \mu \sum_{j=1}^{n} a_{ij} x_j, \quad i = 1, \ldots, n$$

Avec facteur de proportionnalité μ = 1/λ de sorte que xi est proportionnel à la somme des scores de similarité de tous les nœuds qui lui sont connectés.

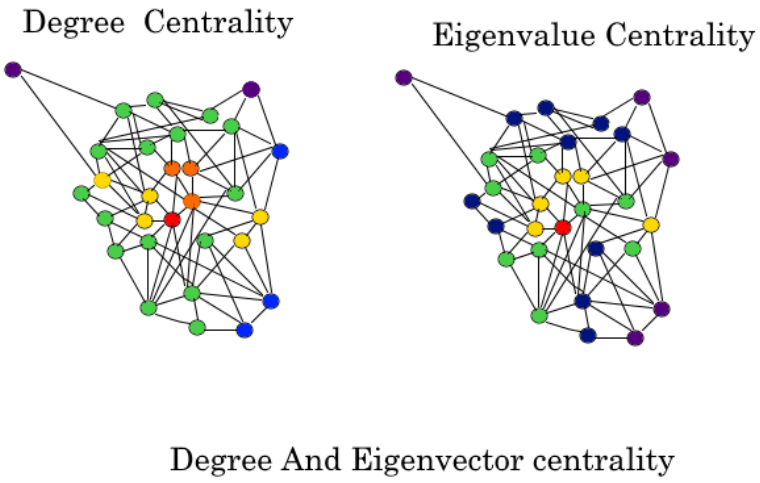

Figure 4.2.4 : La différence entre DC et EC

-Apres avoir définir les mesures de centralités nous allons passer pour présenter le processus de topsis

1. Première étape :

Construction de la matrice d'évaluation qui va contenir dans chaque ligne un model alternative, et les colonnes représente les mesures de centralité que ça soit BC, DC, CC OU EC.

2. Deuxième étape :

Normalisation de la matrice d'adjacence.

3. Troisième étape :

Multiplier les colonnes de la matrice de décision normalisée par les pondérations associées pour obtenir la matrice de décision pondérée.

4. Quatrième étape :

Calculer la distance entre l'alternative cible et l'idéal négatif.

5. Cinquième étape :

Sauvegarder les résultats dans des fichiers. Csv.

Après le passage des différentes étapes qu'on a définies précédemment on a obtenu le tableau suivant qui représente les valeurs de centrality de chaque nœud aussi les valeurs trouvées par la méthode TOPSIS sur le réseau Facebook.

- **Pour le réseau Facebook ego**

	nœud	DC	CC	BC	EC	TOPSIS
0	107	0.258791	0.459699	0.480518	2.606940e-04	0.913277
1	1648	0.196137	0.393606	0.337797	7.164260e-06	0.695566
2	1912	0.186974	0.350947	0.229295	9.540696e-02	0.496063
3	3437	0.135463	0.314413	0.236115	9.531613e-08	0.488865
4	0	0.085934	0.353343	0.146306	3.391796e-05	0.304321
...	...	...	...	...	...	...
4034	775	0.000495	0.178262	0.000000	6.433195e-14	0.000229
4035	749	0.000495	0.178262	0.000000	6.433195e-14	0.000229
4036	841	0.000495	0.178262	0.000000	6.446055e-14	0.000229
4037	692	0.000248	0.178255	0.000000	6.385265e-14	0.000000
4038	801	0.000248	0.178255	0.000000	6.385265e-14	0.000000

Tableau 4.2.1 : Les résultats de DC, CC, BC, EC et TOPSIS de data set Facebook

- **Pour le réseau Email**

	nœud	DC	CC	BC	EC	TOPSIS
0	104	0.062721	0.378216	0.036931	0.229103	0.949770
1	332	0.045936	0.382820	0.039490	0.123657	0.774723
2	22	0.045053	0.381659	0.033463	0.116593	0.719794
3	41	0.045053	0.377585	0.026024	0.131570	0.643457
4	75	0.037986	0.374339	0.030118	0.082444	0.622879
...	...	...	...	...	...	...
1128	1131	0.000883	0.189965	0.000000	0.000040	0.002720
1129	1124	0.000883	0.186154	0.000000	0.000025	0.001392
1130	1130	0.000883	0.185513	0.000000	0.000022	0.001168

1131	1051	0.000883	0.184545	0.000000	0.000030	0.000831
1132	1132	0.000883	0.182169	0.000000	0.000014	0.000000

Tableau 4.2.2 : Les résultats de DC, CC, BC, EC et TOPSIS de data set Email

- **Pour le réseau Football**

	nœud	DC	CC	BC	EC	TOPSIS
0	0	0.105263	0.423792	0.032490	0.106503	0.924560
1	64	0.096491	0.422222	0.033533	0.100915	0.912863
2	106	0.096491	0.435115	0.029161	0.090415	0.819665
3	41	0.087719	0.436782	0.028823	0.079146	0.786272
4	74	0.096491	0.404255	0.025187	0.080489	0.699216
...	...	...	...	...	...	...
110	88	0.087719	0.377483	0.006498	0.070452	0.154306
111	97	0.087719	0.391753	0.005317	0.080728	0.151214
112	91	0.061404	0.370130	0.007790	0.048312	0.149053
113	21	0.087719	0.386441	0.004372	0.081921	0.139444
114	77	0.087719	0.373770	0.003906	0.076032	0.120583

Tableau 4.2.3 : Les résultats de DC, CC, BC, EC et TOPSIS de data set Football

- **Pour le réseau Zachary**

	nœud	DC	CC	BC	EC	TOPSIS
0	0	0.105263	0.423792	0.032490	0.106503	0.924560
1	64	0.096491	0.422222	0.033533	0.100915	0.912863
2	106	0.096491	0.435115	0.029161	0.090415	0.819665
3	41	0.087719	0.436782	0.028823	0.079146	0.786272
4	74	0.096491	0.404255	0.025187	0.080489	0.699216
...	...	...	...	...	...	...
110	88	0.087719	0.377483	0.006498	0.070452	0.154306
111	97	0.087719	0.391753	0.005317	0.080728	0.151214
112	91	0.061404	0.370130	0.007790	0.048312	0.149053
113	21	0.087719	0.386441	0.004372	0.081921	0.139444

114	77	0.087719	0.373770	0.003906	0.076032	0.120583

Tableau 4.2.4 : Les résultats de DC, CC, BC, EC et TOPSIS de data set Zachary

- **Pour le réseau Dolphins**

	nœud	DC	CC	BC	EC	TOPSIS
0	0	0.105263	0.423792	0.032490	0.106503	0.924560
1	64	0.096491	0.422222	0.033533	0.100915	0.912863
2	106	0.096491	0.435115	0.029161	0.090415	0.819665
3	41	0.087719	0.436782	0.028823	0.079146	0.786272
...	...	...	...	...	...	...
110	88	0.087719	0.377483	0.006498	0.070452	0.154306
111	97	0.087719	0.391753	0.005317	0.080728	0.151214
112	91	0.061404	0.370130	0.007790	0.048312	0.149053
113	21	0.087719	0.386441	0.004372	0.081921	0.139444
114	77	0.087719	0.373770	0.003906	0.076032	0.120583

Tableau 4.2.5 : Les résultats de DC, CC, BC, EC et TOPSIS de data set Dolphins

A partir de ces résultats nous avons classé les nœuds chacun par apport à sa valeur de DC, CC, BC, et EC pour les top-10 de notre data set.
Après le classement on a obtenu les résultats suivants :

- **Pour le réseau Facebook ego**

Index	DC	CC	BC	EC	TOPSIS
0	107	107	107	1912	107
1	1684	1684	1684	2266	1684
2	1912	58	1912	2233	1912
3	3437	428	3437	2206	3437
4	0	563	0	2142	0
5	2347	414	1085	2218	1085
6	2543	348	698	2078	698
7	1888	483	567	2464	567
8	1800	376	58	2123	58
9	1663	171	428	1993	428

Tableau 4.2.6 : Les tops 10 de DC, CC, BC, EC et TOPSIS de data set Facebook

- **Pour le réseau Email**

	DC	BC	CC	EC	TOPSIS

0	104	104	104	104	104
1	332	332	332	332	332
2	22	22	22	41	22
3	41	41	41	195	41
4	40	75	75	15	75
5	577	40	40	48	40
6	232	577	232	115	577
7	195	232	134	203	232
8	15	134	377	2	195
9	377	354	51	55	134

Tableau 4.2.7 : Les tops 10 de DC, CC, BC, EC et TOPSIS de data set Email

- **Pour le réseau Football**

	BC	CC	EC	TOPSIS
0	0	0	28	0
1	64	106	114	64
2	106	41	103	106
3	41	80	52	41
4	74	4	82	74
5	46	28	101	46
6	80	114	81	80
7	24	35	108	24
8	83	25	61	83
9	4	67	113	4

Tableau 4.2.8 : Les tops 10 de DC, CC, BC, EC et TOPSIS de data set Football

- **Pour le réseau Zachary**

	DC	CC	EC	BC	TOPSIS
0	0	0	0	0	0
1	33	33	33	33	33
2	32	32	32	32	32
3	2	2	2	2	2
4	1	31	31	31	31
5	8	1	1	1	1
6	13	8	8	8	8
7	23	13	13	13	13

8	9	3	3	19	3
9	22	19	30	27	19

Tableau 4.2.9 : Les tops 10 de DC, CC, BC, EC et TOPSIS de data set Zachary

- **Pour le réseau Dolphins**

index	DC	BC	CC	EC	TOPSIS
0	36	36	36	37	36
1	14	1	1	40	1
2	54	37	37	14	37
3	43	40	40	51	40
4	50	20	20	33	20
5	15	51	14	45	14
6	18	17	7	29	51
7	52	7	33	50	17
8	9	54	28	21	7
9	44	57	8	16	33

Tableau 4.2.10 : Les tops 10 de DC, CC, BC, EC et TOPSIS de data set Dolphins

4.3. Application de K-means

L'objectif de cette partie est de faire le clustring les nœuds du dataset qu'on a déjà défini précédemment. Le travail détaillé sera fait sur le réseau Facebook ego et les résultats des autres réseaux nous allons l'exprimer sous la forme des tableaux qui vont contenir les centroides de chaque cluster.
On implémente k-means premièrement avec la matrice de décision et deuxièmement avec la matrice d'adjacence.

- **k-means avec la matrice de décision**

1. On applique le k-mean sur la matrice de décision qui contient les valeurs des mesures des centralités résultats d'application de la méthode TOPSIS.
2. On initialise l'algorithme avec les top-10, comme l'étalement des centroïdes initiaux est considéré comme un objectif louable, k-means poursuit en assignant les centroïdes initiaux à l'emplacement des nœuds influents sélectionnés dans la méthode proposée, puis choisir les centroïdes suivants parmi les points de données restants en fonction

d’une probabilité proportionnelle à la distance au carré à partir d’un centroïde existant le plus proche.

3. Construction du graphe qui montre les regroupements résultats de l’algorithme k-means.

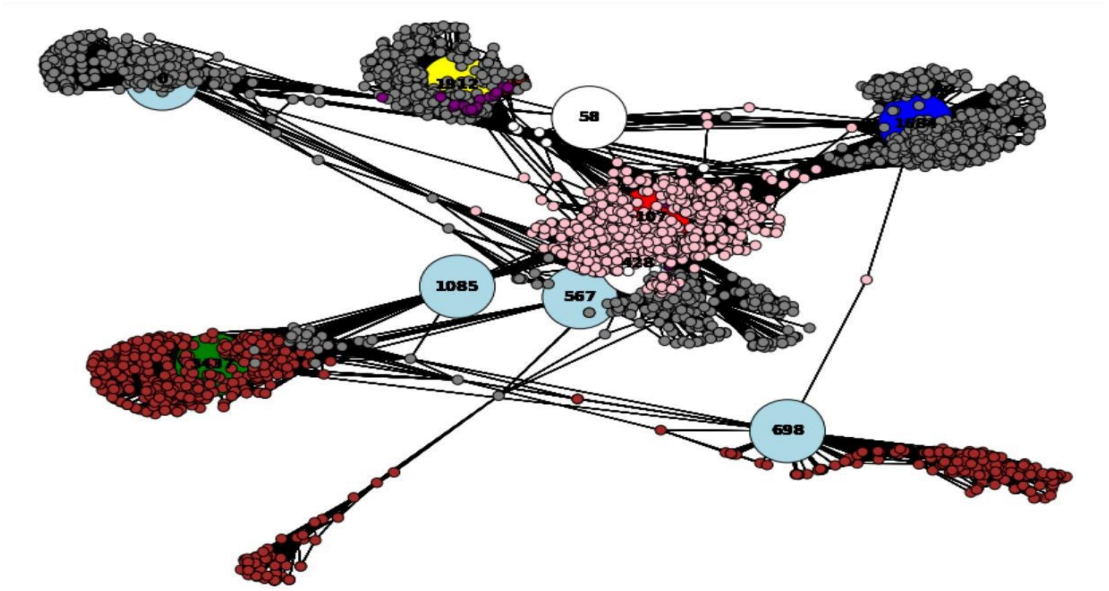

Figure 4.3.1 : Graphe de l’algorithme K-Means avec la matrice de décision

Nous remarquons que nous avons 3 couleurs dominantes dans le graphique. Ainsi, nous avons 3 communautés principales Concernant les nœuds influents découverts par la méthode proposée.

Nous remarquons que nous avons des communautés qui contiennent deux (ou plus) nœuds influents (exemple, communauté 4 avec la couleur cyan et 5 avec la couleur blanche), Et les autres communautés (0, 1,2 et 3) contiennent un nœud :

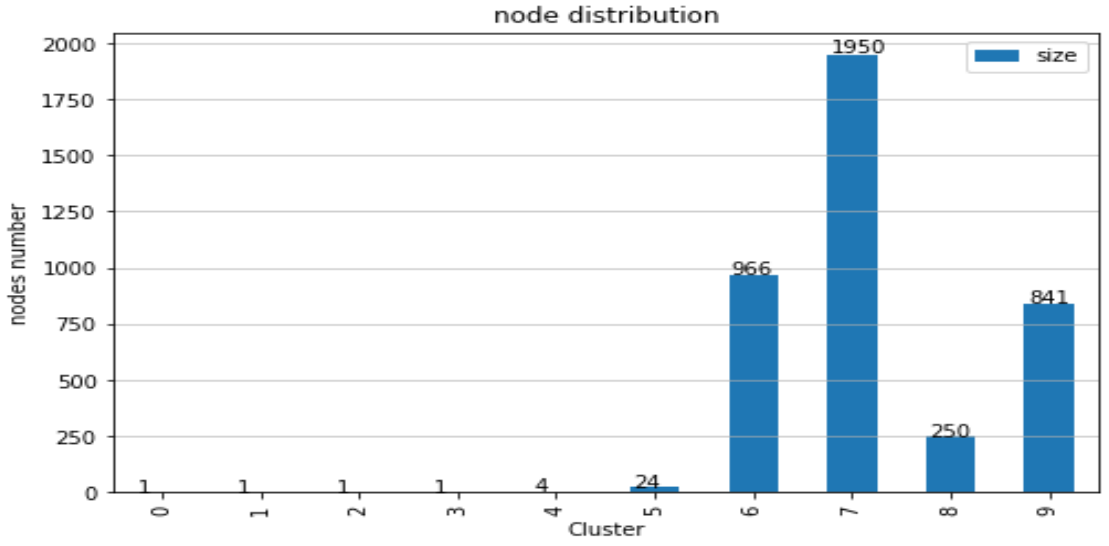

Figure 4.3.2 : Le nombre cumulative des nœuds pour chaque cluster

Le graphe suivant montre que la répartition des nœuds dans chaque communauté n'est pas bien répartie : la communauté 8, 7 et 9 contient la majorité des nœuds, tandis que les autres communautés partagent le reste des nœuds avec de faibles pourcentages.

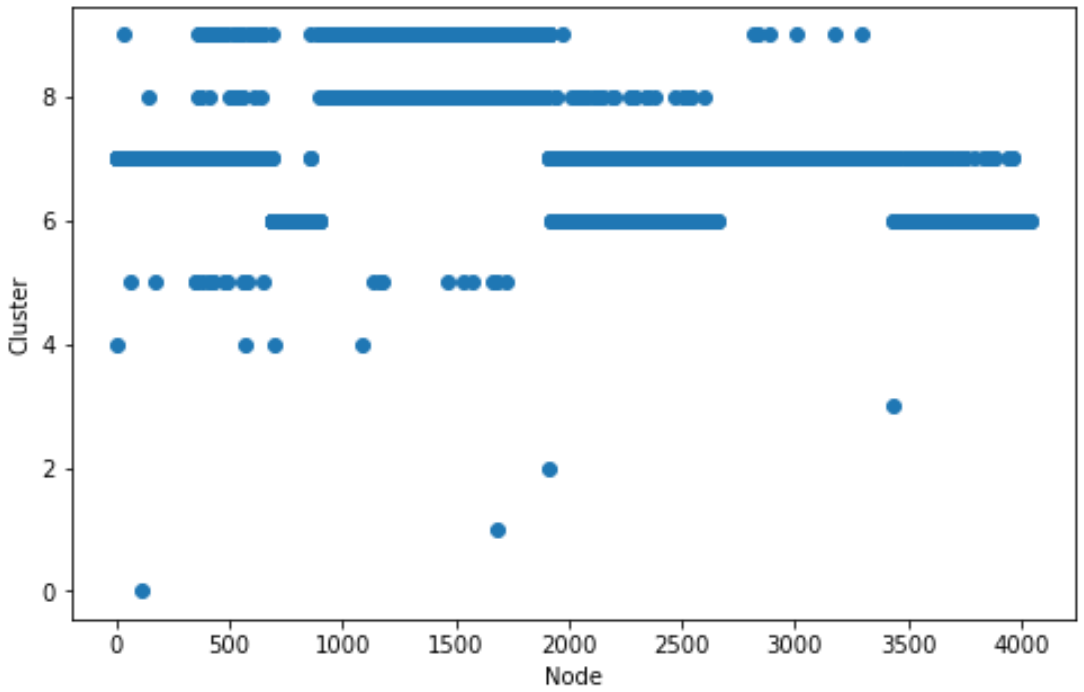

Figure 4.3.3 : Diagramme des communite K-means base sur la matrice de décision

- **k-mean avec la matrice d'adjacence**

On va suivre la même procédure mais cette fois on se base sur la matrice d'adjacence. Le résultat de l'implémentation est le suivant :

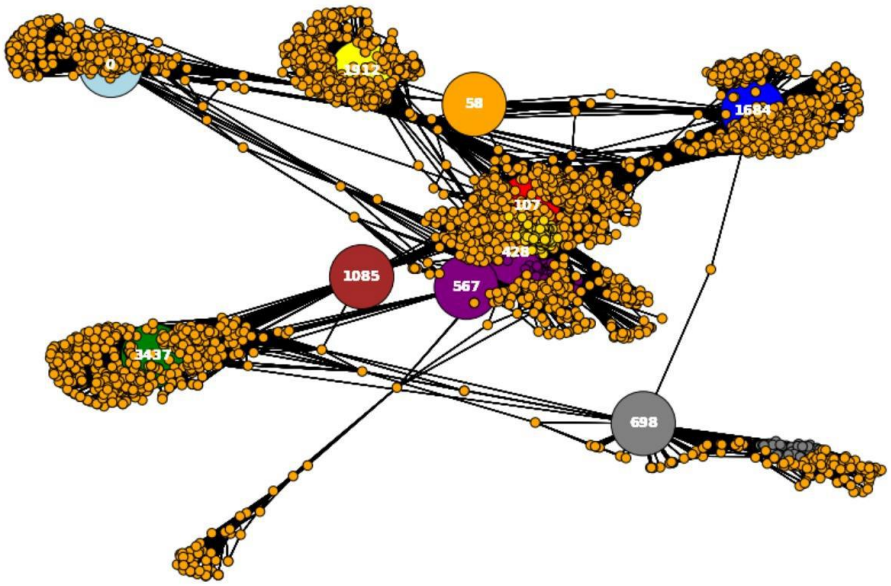

Figure 4.3.4 : Graphe de l'algorithme K-means avec la matrice d'adjacence

La seule communauté dominante dans le graphique est celle avec la couleur orange, le nœud 58 appartient à cette communauté, mais il semble que les autres nœuds sont affectés à beaucoup d'autres communautés. Ainsi, nous pouvons considérer que la matrice d'adjacence ne convient pas pour être utilisée pour détecter les communautés dans une telle situation.

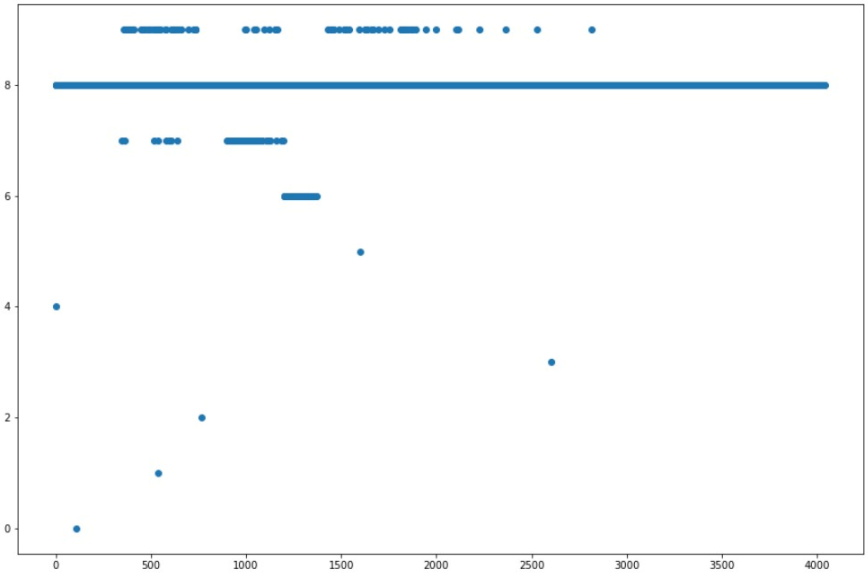

Figure 4.3.5 : Diagramme des communite K-means base sur la matrice d'adjacence

Comme on peut le voir, le graphe montre comment la répartition des nœuds dans chaque communauté n'est pas bien répartie : Communauté 8 et 9 contiennent la majorité des nœuds, tandis que les autres communautés partagent le reste des nœuds avec de faibles pourcentages et c'est ça qui justifie les résultats montré dans le précèdent graphe.

- **Pour le réseau Facebook ego**

	cluster	Node_avant_KMeans	Node_ apres _KMeans	distance from center
0	0	107	107.0	0.000000e+00
1	1	1684	1684.0	1.270549e-19
2	2	1912	1912.0	0.000000e+00
3	3	3437	3437.0	6.950964e-20
4	4	0	567.0	3.541115e-02
5	5	1085	1465.0	1.425377e-02
6	6	698	3758.0	1.678485e-02
7	7	567	487.0	8.161578e-04
8	8	58	1456.0	2.899335e-03
9	9	428	1543.0	1.700190e-04

Tableau 4.3.1 : Les distances entre les nœuds avant k_means et après le k_means de data set Facebook ego

Nous avons remarqué que, selon la figure ci-dessus, les trois premiers centroïdes initiaux sont les mêmes après l'application des K-means. Pour le reste des nœuds la distance entre eux et les centroïdes finaux est importante.

- **Pour le réseau Email**

	cluster	Node_avant_KMeans	Node_ apres _KMeans	distance from center
0	0	104	104	0.000000
1	1	332	53	0.013519
2	2	22	181	0.002979
3	3	41	321	0.004551
4	4	75	239	0.001688
5	5	40	742	0.001411
6	6	577	501	0.000566
7	7	232	718	0.001162
8	8	195	55	0.007047
9	9	134	1010	0.000847

Tableau 4.3.2 : Les distances entre les nœuds avant k_ means et après le k_means de data set Email

Nous avons remarqué que, selon la figure ci-dessus, le premiers centroïdes initial est le mêmes après l'application des K-means. Pour le reste des nœuds la distance entre eux et les centroïdes finaux est diminué à partir de deuxième centroïde.

- **Pour le réseau Football**

	cluster	Node_avant_KMeans	Node_ apres _KMeans	distance from center
0	0	0	28.0	0.002301
1	1	64	4.0	0.003756
2	2	106	80.0	0.003754
3	3	41	67.0	0.007873
4	4	74	49.0	0.005705
5	5	46	9.0	0.005111
6	6	80	94.0	0.003046
7	7	24	59.0	0.007550
8	8	83	104.0	0.003952
9	9	4	34.0	0.004087

Tableau 4.3.3 : Les distances entre les nœuds avant k_ means et après le k_means de data set Football

Nous avons remarqué que, selon la figure ci-dessus, les centroïdes initiaux ne sont pas les mêmes après l'application des K-means. Mais la distance entre eux et les centroïdes finaux est faible.

- **Pour le réseau Zachary**

	cluster	Node_avant_KMeans	Node_ apres _KMeans	distance from center
0	0	0	0	0.000000
1	1	33	33	0.000000
2	2	32	32	0.000000
3	3	2	2	0.000000
4	4	31	31	0.000000
5	5	1	1	0.000000
6	6	8	8	0.005053
7	7	13	3	0.000000
8	8	3	27	0.025601
9	9	19	12	0.012871

Tableau 4.3.4 : Les distances entre les nœuds avant k_ means et après le k_means de data set Zachary

Nous avons remarqué que, selon la figure ci-dessus, les sept premiers centroïdes initiaux sont les mêmes après l'application des K-means. Pour le reste des nœuds la distance entre eux et les centroïdes finaux est importante.

- **Pour le réseau Dolphins**

	cluster	Node_avant_KMeans	Node_ apres _KMeans	distance from center
0	0	36	36.0	0.000000
1	1	1	54.0	0.030748
2	2	37	37.0	0.000000
3	3	40	40.0	0.025614
4	4	20	0.0	0.014418
5	5	14	14.0	0.000000
6	6	51	38.0	0.015231
7	7	17	49.0	0.007472
8	8	7	30.0	0.015635
9	9	33	45.0	0.015042

Tableau 4.3.5 : Les distances entre les nœuds avant k_ means et après le k_means de data set Dolphins

Nous avons remarqué que, selon la figure ci-dessus, le premiers, le troisième, le quatrième, et le cinquième centroïdes initiaux sont les mêmes après l'application du k-means. Pour le reste des nœuds la distance entre eux et les centroïdes finaux est importante.

4.4. Discussions

La discussion des résultats sera basée sur le model SI qui va évaluer le rendement de notre méthode de classement, nous utilisons le modèle Sensible-Infecté (SI) pour examiner l'influence de propagation des nœuds les mieux classés. Il a été largement utilisé pour la dynamique épidémique sur les réseaux. Dans le modèle SI, chaque nœud a deux états distincts : (i) Sensible S(t) représente le nombre d'individus sensibles à la maladie (pas encore infectés); (ii) Infecté I(t) indique le nombre d'individus qui ont été infectés et qui sont en mesure de transmettre la maladie à des individus sensibles. À chaque étape, pour chaque nœud infecté, un voisin susceptible au hasard est infecté avec la probabilité λ (pour l'uniformité, ici nous fixons λ = 0,3 initialement). Pour la propagation épidémique sur les réseaux, λ détermine la portée ou l'échelle sur laquelle un nœud peut exercer une influence.
La propagation épidémique sur les réseaux est un bon exemple pour illustrer le concept multi-échelle : un nœud infecté peut propager la maladie non seulement à ses voisins immédiats, mais aussi à ses voisins d'ordre supérieur à travers les intermédiaires. Dans ce mode, le nombre de nœuds infectés au moment t est indiqué par F (t).
À l'état d'équilibre (état final), le nombre de nœuds infectés est identique en utilisant différents nœuds initialement infectés, et il devrait être égal au nombre total de nœuds dans les réseaux. Moins le réseau atteint l'état stable, plus le nœud initialement infecté est influent.
Cela signifie que l'indicateur pour évaluer l'influence du nœud initialement infecté est le nombre moyen de nœuds infectés à chaque étape ou le taux de propagation. Après avoir définir la notion de Si modèle nous allons l'applique sur le grand réseau Facebook ego, et Email après sur les petits réseaux. Nous évaluons les résultats obtenus par la méthode TOPSIS, et les centroïdes des clusters de K-means. Dans un premier temps, nous comparons la méthode proposée avec DC, CC, BC et EC. Après nous comparons entre les mesures de centrality et les centroïdes après l'application de l'algorithme de k-means, et finalement en à faire une comparaison entre le nœud quand à utiliser pour initialiser l'algorithme de K-mean et les centroïdes finale de lui.

1. La comparaison entre les mesures de centralité et la méthode proposée par TOPSIS

DC, CC, BC et EC sont comparés à la méthode proposée (TOPSIS). Simultanément, dans Facebook ego Network dans lequel l'information circule au moyen de promenades, nous comparons la méthode proposée avec DC, CC, BC et EC. Si l'on compare la méthode proposée avec celle de DC, on constate qu'il y a cinq mêmes membres dans les 10 premières listes. Donc le nœud 107 est le nœud le plus influent dans le réseau de l'ego de Facebook en utilisant ces quatre mesures. Cette comparaison est à la base du la figure ci-dessous.
Maintenant on va dessiner le graphe qui va représenter Le nombre cumulatif de nœud infectés en fonction du temps, les nœuds initialement infectés étant ceux qui apparaissent dans la liste des 10 premiers par la méthode proposée ou DC, CC, et EC. Les résultats sont obtenus par une moyenne de plus de 100 000 implémentations (λ=0.3), et sont présenter dans la figure suivant

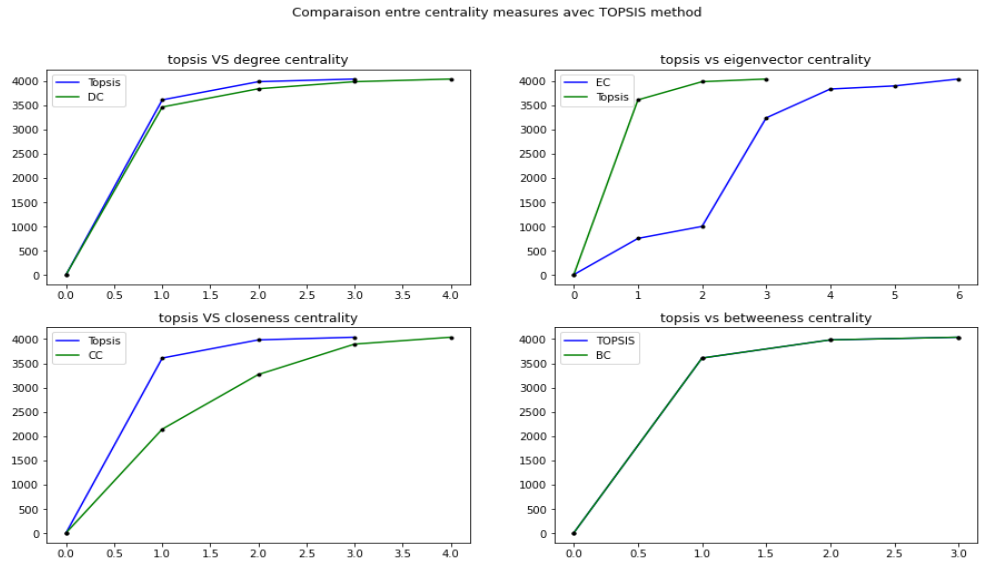

Figure 4.4.1 : Diagramme de nombre cumule des nœuds infects en fonction de temps les nœuds initialement Infecte étant ceux qui apparaissaient à la liste de top 10 de réseau Facebook ego

Depuis la figure ci-dessus TOPSIS et la betweenes centralité ont le même top 10 nœuds influents : Ce ne sont pas seulement les utilisateurs qui ont le plus d'amis qui sont importants, les utilisateurs qui relient une géographie à l'autre sont également importants car cela permet aux utilisateurs de voir le contenu de diverses géographies. Entre temps, la centralité quantifie le nombre de fois qu'un nœud particulier arrive dans le chemin le plus court choisi entre deux autres nœuds.

2. La comparaison entre les mesures de centralité et centroïde de l'algorithme k-means :

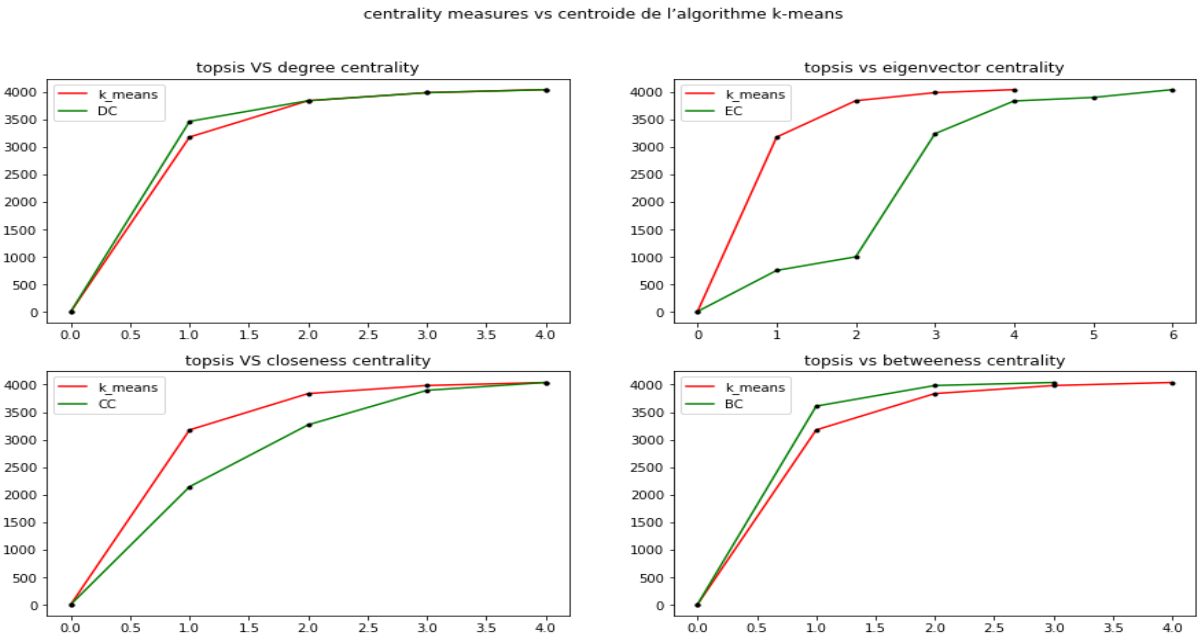

Figure 4.4.2 : Diagramme de nombre cumule des nœuds infects en fonction de temps Le nœud initialement infect étant ceux qui apparaissaient à la liste des centroide De K-means, appliqué sur le réseau Facebook ego

D’après la figure ci-dessus ; On trouve que l’algorithme k-means est un peu meilleur qu’EC et CC pour le nombre moyen de nœuds infectés à chaque étape. Nous pouvons également voir que DC et BC surpasse les centroïde après k_means.
La méthode proposée nécessite moins d'étapes pour atteindre l'état d'équilibre que DC et EC.
Alors k-means n’a pas les meilleures performances que les quatre autres mesures de centralité.

3. La comparaison entre les centroïdes de l’algorithme k-means et la méthode proposée par TOPSIS :

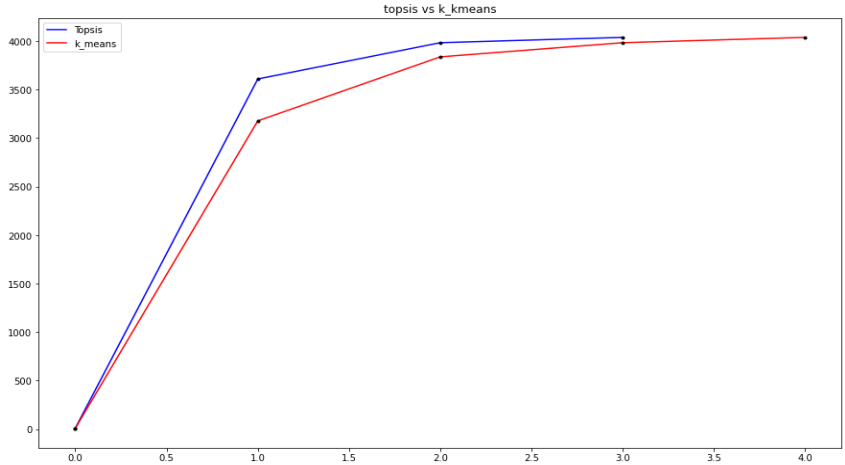

Figure 4.4.3 : Diagramme de nombre cumule des nœuds infecte en fonction de temps les nœuds initialement infecte étant ceux qui apparaissait au liste des top 10 et les centroide de K-means .Appliquer sur le réseau Facebook ego

D’après la figure ci-dessus Le nombre des nœuds infectés cumulés augmente avec le temps et atteint finalement la valeur stable. Par la méthode proposée (topsis) et k_means on trouve que la méthode proposée (topsis) est légèrement meilleure que k_means pour le nombre moyen de nœuds infectés à chaque étape.

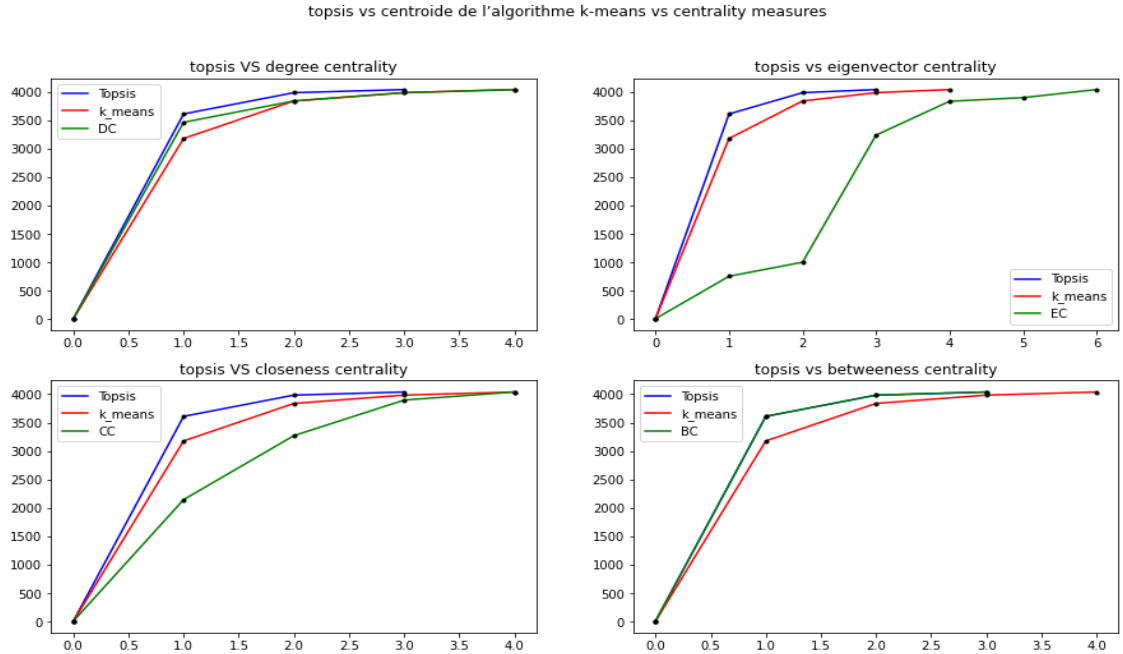

Figure 4.4.4 : Diagramme de nombre cumule des nœuds infecte en fonction de temps les nœuds initialement infecte étant ceux qui apparaissait au liste des top 10 avec TOPSIS , les top 10 avec les mesures de centralités ,et les centroide de K-means ,Appliquer sur le réseau Facebook ego

- **Pour le réseau Email**

Les différents diagrammes de l'implémentation sont les suivant :

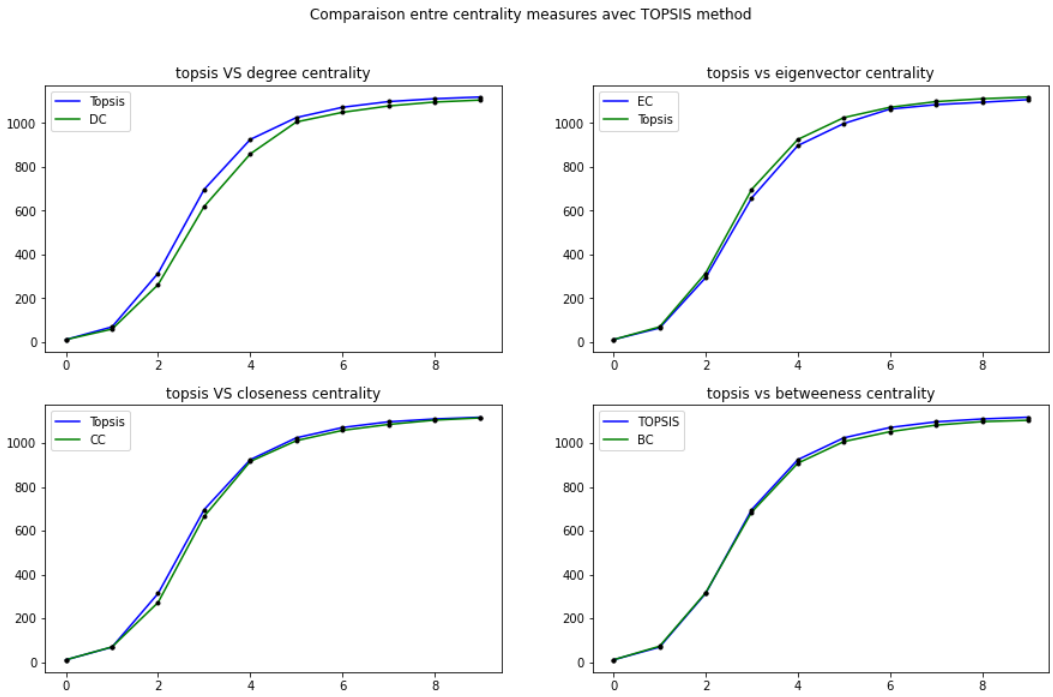

Figure 4.4.4 : Diagramme de nombre cumule des nœud infecte en fonction de temps les nœuds initialement infecte étant ceux qui apparaissait au liste des top 10 avec TOPSIS ,et les top 10 avec les mesures de centralités ,Appliquer sur le réseau Email

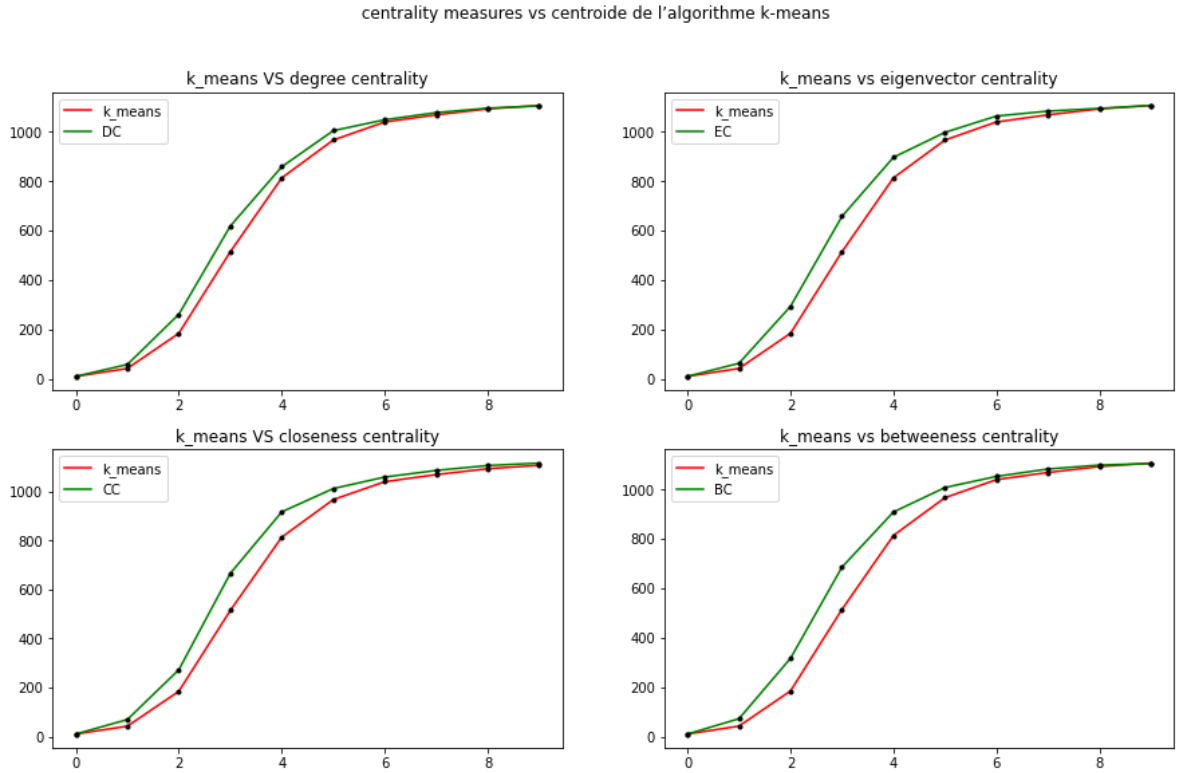

Figure 4.4.5 : Diagramme de nombre cumule des nœud infecte en fonction de temps les nœuds initialement infecte étant ceux qui apparaissait au liste des centroide de K-means ,et les top 10 avec les mesures de centralités ,Appliquer sur le réseau Email

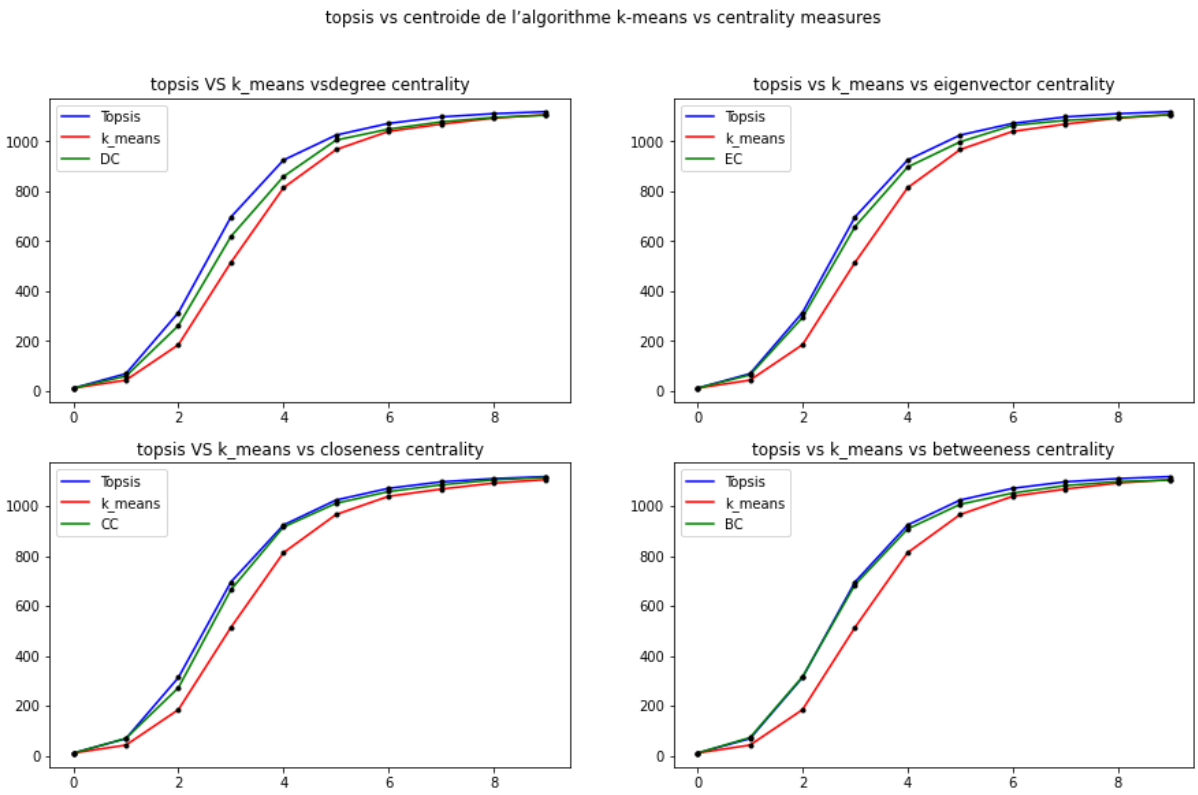

Figure 4.4.6 : Diagramme de nombre cumule des nœud infecte en fonction de temps les nœuds initialement infecte étant ceux qui apparaissait au liste des top 10 avec TOPSIS , les top 10 avec les mesures de centralités ,et les centroide de K-means appliquer sur le réseau email

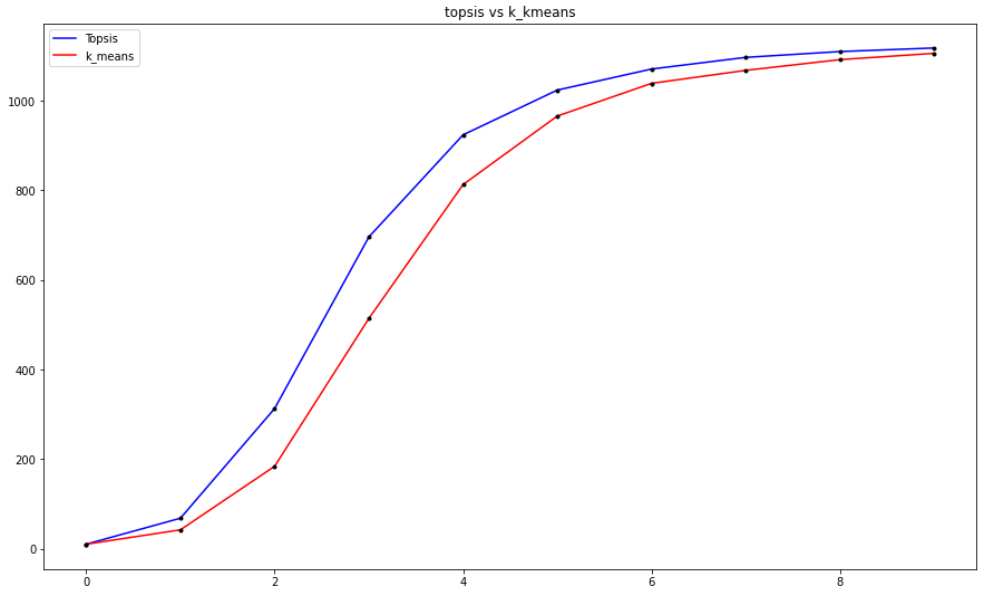

Figure 4.4.6 : Diagramme de nombre cumule des nœud infecte en fonction de temps les nœuds initialement infecte étant ceux qui apparaissait au liste des top 10 avec TOPSIS , les top 10 avec les mesures de centralités ,et les centroide de K-means, appliquer sur le réseau Email

A partir des diagrammes en voix clairement que la méthode topsis est le plus adapter à ces réseaux et ça se voit à partir de premiers diagramme qui montrent la compatibilité de toutes les mesures de centrality avec la méthode proposée par la méthode de topsis donc en peu dire qu'ils sont les mêmes tops 10.

Le deuxième diagramme montre que les centroïdes de k-mean et moins infecter par apport aux nœuds des mesures de centrality.

Donc pour identifier les influenceurs de ces réseaux en peut se base sur la méthode de topsis.

- **Pour le réseau Zachary**

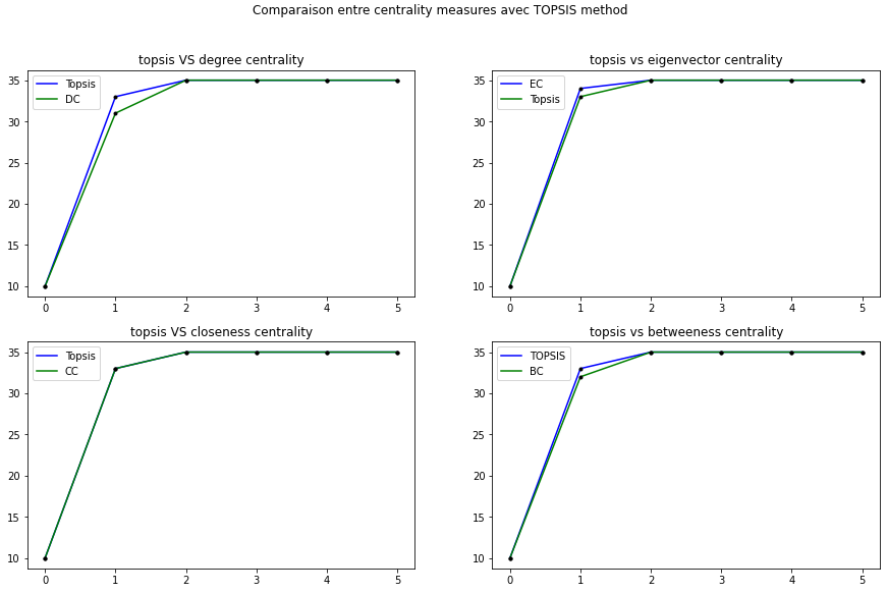

Figure 4.4.8 : Diagramme de nombre cumule des nœud infecte en fonction de temps les nœuds initialement infecte étant ceux qui apparaissait au liste des top 10 avec TOPSIS ,et les top 10 avec les mesures de centralités ,Appliquer sur le réseau Zachary

On a TOPSIS et CC (Closeness centrality) ont le même top 10 nœuds influents, les utilisateurs qui relient une géographie à l'autre sont également importants. Entretemps, la centralité quantifie le nombre de fois qu'un nœud particulier arrive dans le chemin le plus court choisi entre deux autres nœuds.

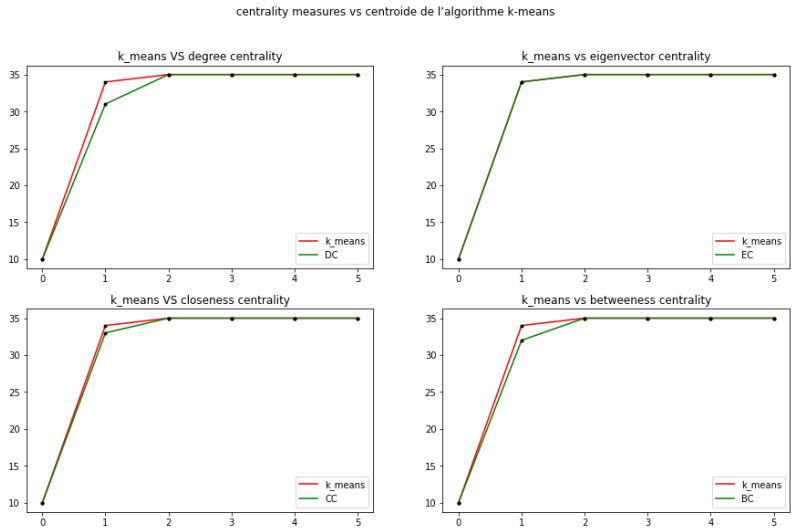

Figure 4.4.9 : Diagramme de nombre cumule des nœud infecte en fonction de temps les nœuds initialement infecte étant ceux qui apparaissait au liste des centroide de K-means ,et les top 10 avec les mesures de centralités ,Appliquer sur le réseau Zachary

Cette fois ci on voie clairement que la courbe d'EC (eigenvector centrality) et k_means en les mêmes tops 10. Dans les autres diagrammes celles de k-mean vs DC et k-mean vs CC

et k-mean vs BC les centroïdes de k mean dans la première itération infecte un nombre des nœuds plus grand que celui infecter par les mesure de centrality.
Comme le diagramme le montre toujours les centroïdes des k-mean infecte le maximum des nœuds dans la première itération suivie par les nœuds de topsis et à la fin les courbes de mesures de centralités si les top10 des mesures et de topsis n'est pas les mêmes.

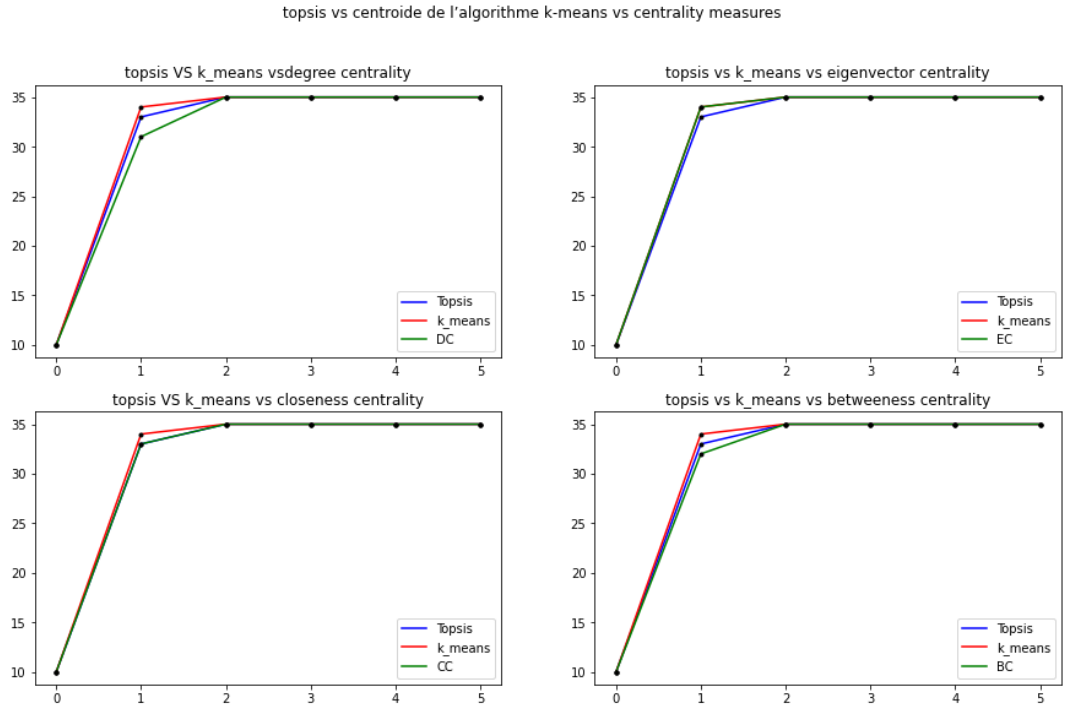

Figure 4.4.10 : Diagramme de nombre cumule des nœud infecte en fonction de temps les nœuds initialement infecte étant ceux qui apparaissait au liste des top 10 avec TOPSIS , les top 10 avec les mesures de centralités ,et les centroide de K-means ,Appliquer sur le réseau Zachary

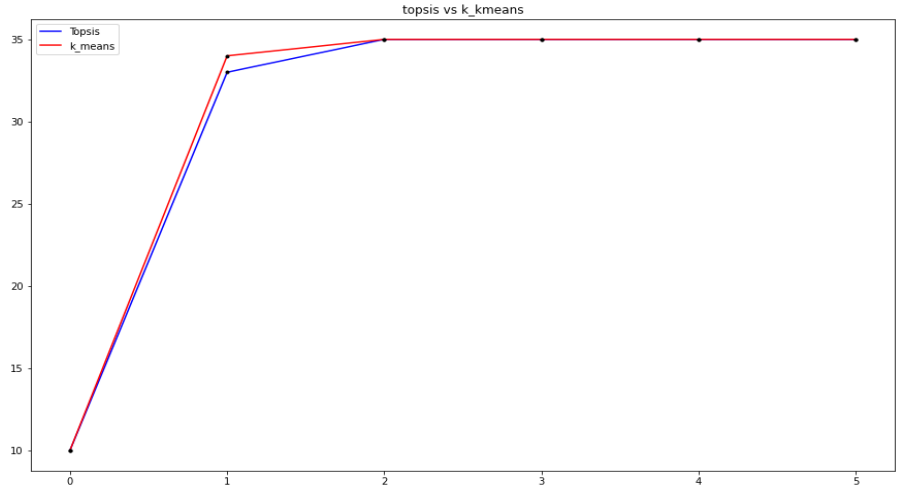

Figure 4.4.11 : Diagramme de nombre cumule des nœuds infects en fonction de temps les nœuds initialement infecte étant ceux qui apparaissait au liste des top 10 et les centroide de K-means .Appliquer sur le réseau Zachary

Le diagramme suivant montre la quatrième partie de notre analyse a ce réseau, il va présenter les centroïdes des k-mean et les tops 10 obtenue par la méthode de topsis, entremet dit est un diagramme qui va présenter les nœuds avec laquelle on a initialisé l'algorithme de k-mean et les nœuds qui ont résulté avec cette algorithme ; si la distance

entre les nœuds d'initialisation et les nœuds résulte est faible en peux considérer les tops 10 comme des centroïdes
D'après le diagramme on peut considérer les centroïdes de k-mean comme des influences car la courbe de k-mean atteindre le nombre maximal des nœuds le premier.

- **Pour le réseau football**

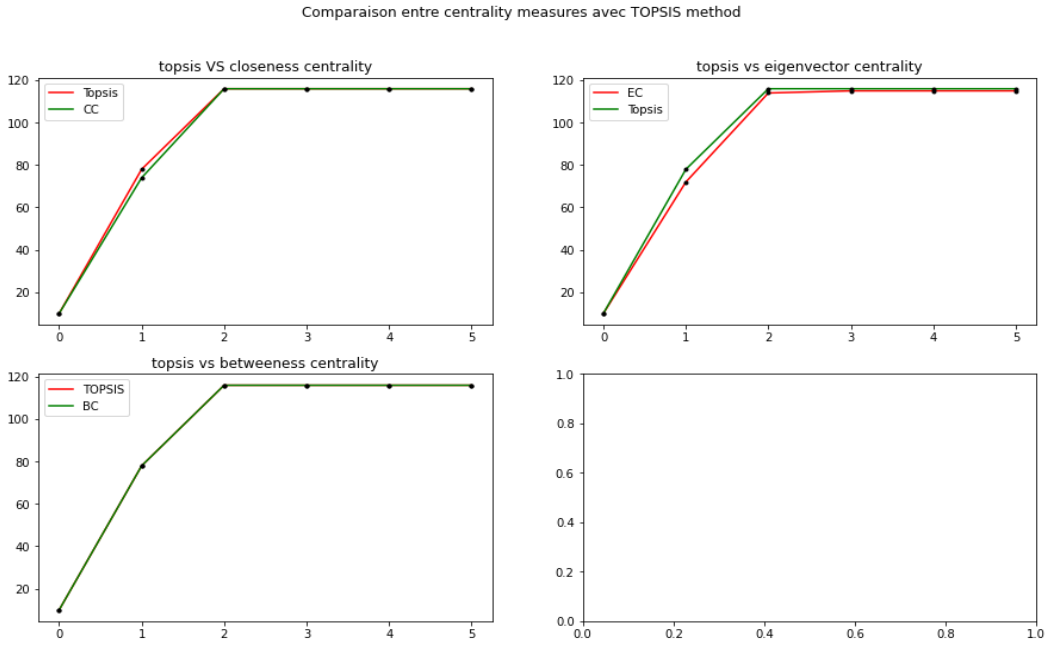

Figure 4.4.12 : Diagramme de nombre cumule des nœuds infecte en fonction de temps les nœuds initialement infecte étant ceux qui apparaissait au liste des top 10 avec TOPSIS ,et les top 10 avec les mesures de centralités ,Appliquer sur le réseau Football

Topsis et BC (beetweeness centrality) ont les tops 10 ce qui explique les résultats obtenus à partir de diagramme topsis vs betweeness centrality, d'où cette data set confort plus avec le BC (beetweness centrality).

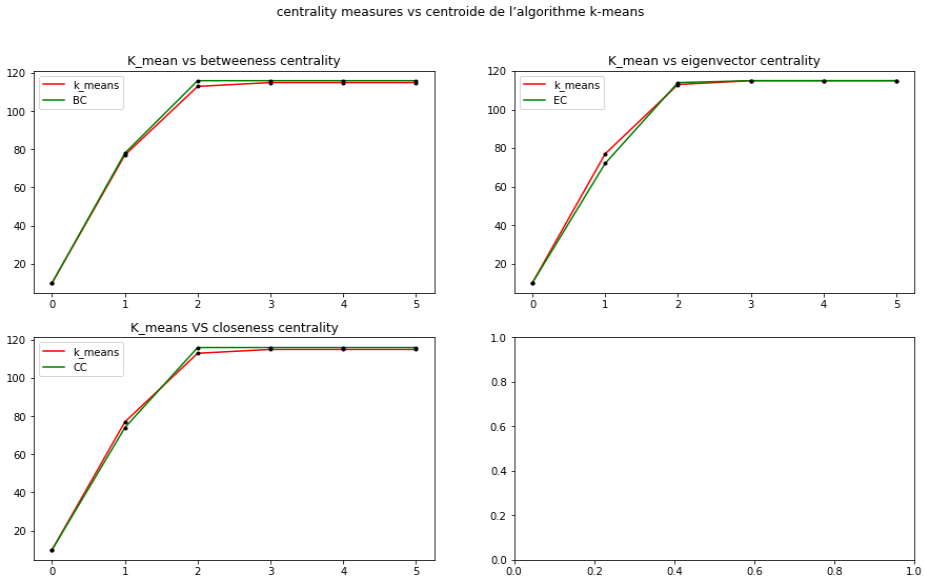

Figure 4.4.13 : Diagramme de nombre cumule des nœuds infecte en fonction de temps les nœuds initialement infecte étant ceux qui apparaissait au liste des centroide de K-means ,et les top 10 avec les mesures de centralités ,Appliquer sur le réseau Football

A partir de diagramme ci-dessus les courbe sont très proche donc les nœuds résultent par le k-mean est adapté aux mesure de centralité, d'où on peut admettre l'algorithme de k-mean comme une méthode pour détecter les influenceurs dans ces réseaux.

4.5. Discussion des résultats de k-mean avec les scores

Dans cette partie nous allons discuter les résultats obtenus par le changement de la fonction qui calcule les distances de l'algorithme k-mean.
La nouvelle fonction il va calculer les scores des nœuds à l'aide des valeurs de mesure de centralité de chaque nœud, nous calculons d'abord la somme des mesures de centralité DC BC CC et EC et nous le divisons sur 4 ; la valeur obtenue on va l'utiliser pour regrouper les nœuds. Apres l'implémentation se fait sur les data sets football, Email, Zachary, et Dolphins.
Pour la data set Facebook ego qui se classe parmi les grand data sets nous utilisons la somme des DC BC et CC diviser sur 3 pour calculer les scores des nœuds.
La comparaison ce fait a l'aide de SI modèle qui va évaluer les nœuds qui résultent avec l'algorithme de k-means et les nœuds qui résulte avec l'algorithme k-means qu'on nous avons développé, et dans le même diagramme on va tracer la courbe des tops 10 da chaque réseaux.

A. Pour Football data set

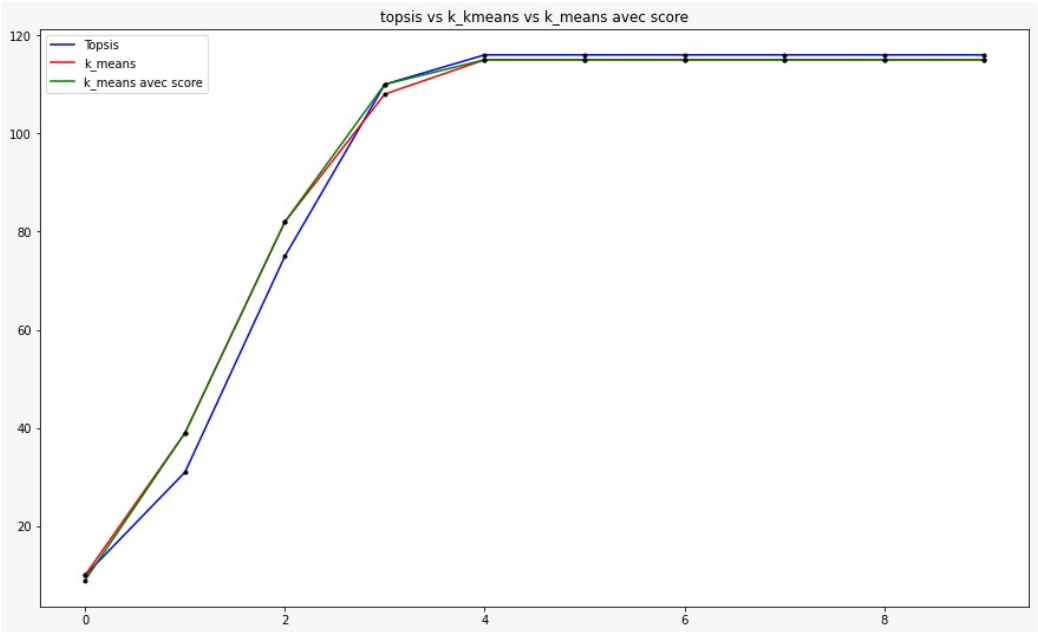

Figure 4.5.a : Diagramme de nombre cumule des nœuds infects en fonction de temps les nœuds initialement infecte étant ceux qui apparaissait à la liste du top 10 avec TOPSIS, les centroide de K-means avec score, et les centroide de K-means, Appliqué sur Football

Nous remarquons que les courbes de k-means et de k-means avec les scores dans le premier et la deuxième itération infecte un nombre maximum des nœuds par rapport au nœud de topsis, après dans la troisième itération les nœuds de topsis et k-means avec score infectent au même temps le nombre maximal des nœuds.

Donc l'utilisation de la méthode topsis ou le k-means avec le score est efficace pour identifier l'influence dans ce réseau.

B. Pour Email data set

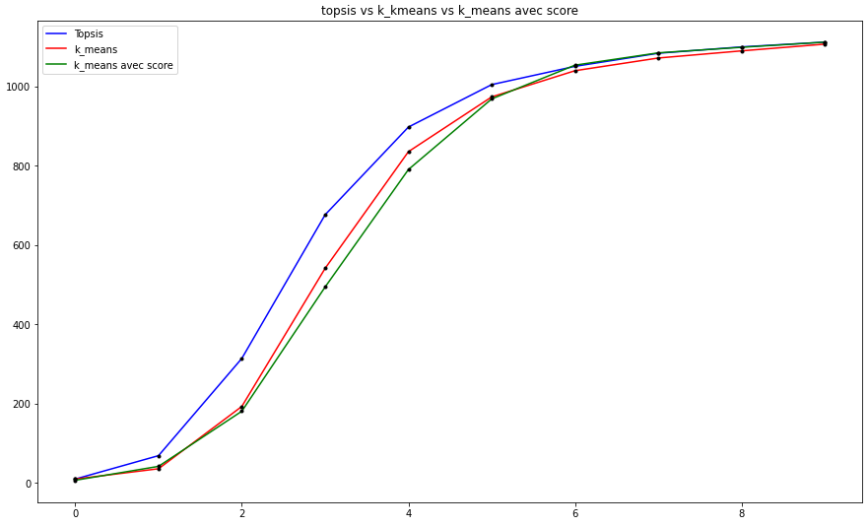

Figure 4.5.b : Diagramme de nombre cumule des nœuds infects en fonction de temps les nœuds initialement infecte étant ceux qui apparaissait à la liste du top 10 avec TOPSIS, les centroide de K-means avec score, et les centroide de K-means, Appliqué sur Email

Pour cette data set on voit que la performance de la méthode de topsis est plus élevée car la courbe de lui est en haut des autre courbes à chaque itération ; les nœuds qui sont choisie parmi les tops 10 de la méthode topsis pour initialiser le SI model infecte le nombre maximale des nœuds de data set en premiers. L'utilisation de k-means avec score n'a pas fait une grand différence sa courbe et la courbe de k-mean sont un peu près les mêmes. Donc pour identifier les influenceurs dans ce réseau il est favorable d'utiliser la méthode de topsis.

C. Pour Zachary data set

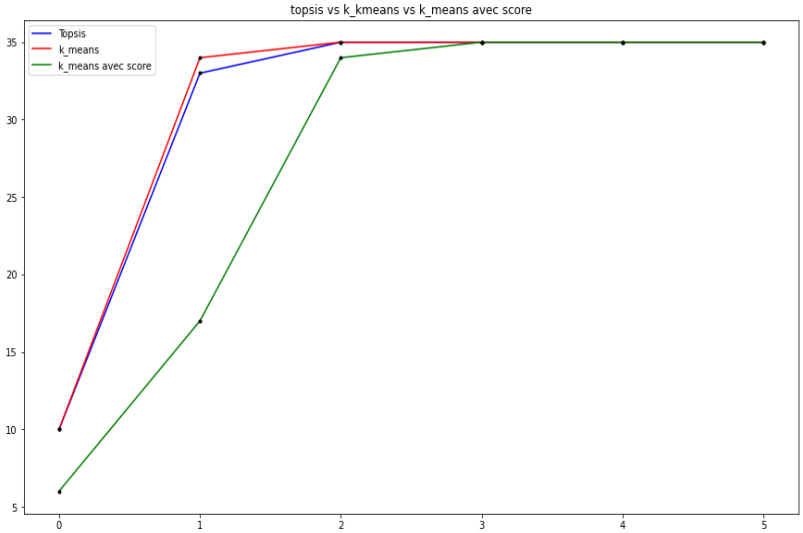

Figure 4.5.c : Diagramme de nombre cumule des nœuds infecte en fonction de temps les nœuds initialement infecte étant ceux qui apparaissait la liste des tops 10 avec TOPSIS, les centroide de K-means avec score, et les centroide de K-means, Appliquer sur le réseau Zachary

Dans cette data set la courbe de k-mean est per fermenté par rapport aux autres ; le k-means avec score n'a pas atteindre les résultats souhaiter sa courbe reste toujours au-dessous que ça soit topsis que ça soit k-means.

D. Pour Dolphins data set

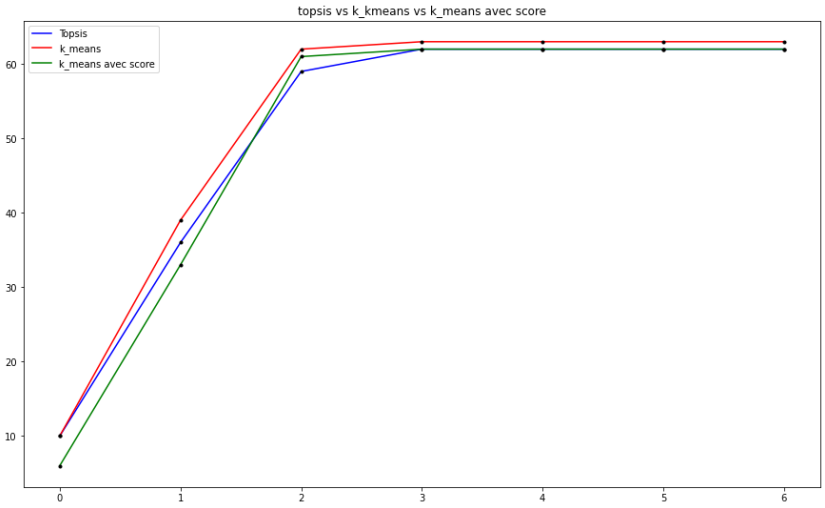

Figure 4.5.d : Diagramme de nombre cumule des nœuds infecte en fonction de temps les nœuds initialement infecte étant ceux qui apparaissait la liste des tops 10 avec TOPSIS, les centroide de K-means avec score, et les centroide de K-means, Appliquer sur le réseau Dolphins

Dans cette exemple nous remarquons que la courbe de k-means avec score est régulière jusqu'à l'infectassions de tous les nœuds, l'utilisation de c'est score à prendre juste deux itération et il a atteindre le nombre maximale des nœuds infecter. Par contre les autres méthodes ont estimé trois itérations pour être dans un état d'équilibre.

E. Pour Facebook ego data set

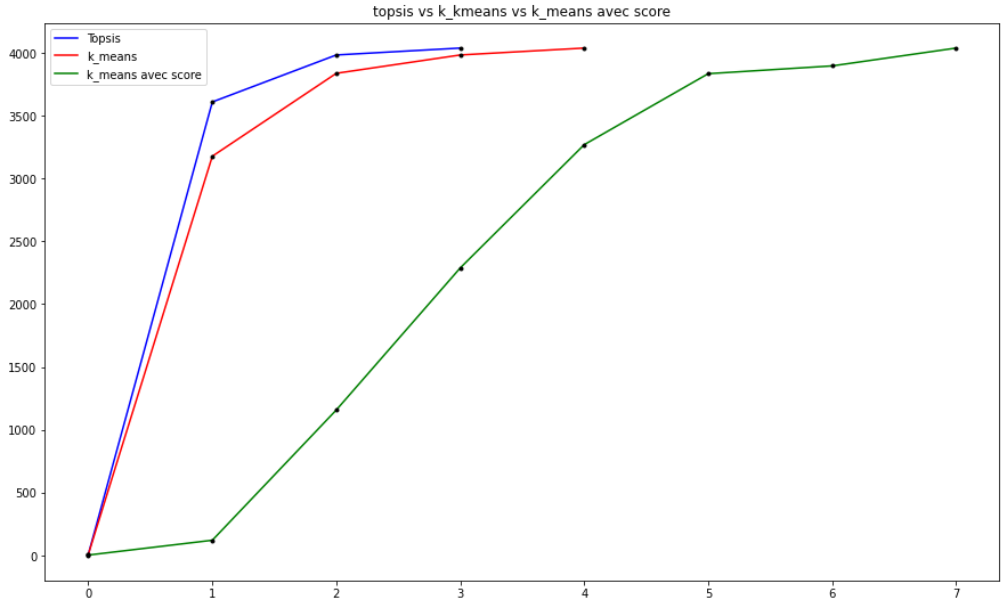

Figure 4.5.e : Diagramme de nombre cumule des nœuds infecte en fonction de temps les nœuds initialement infecte étant ceux qui apparaissait la liste des tops 10 avec TOPSIS, les centroide de K-means avec score, et les centroide de K-means, Appliquer sur Facebook ego

Comme il est montré dans le diagramme les résultats obtenus par k-means avec score à donner des résultats non performants car il a estimé un plus grand nombre d'itération pour avoir un état d'équilibre et aussi pour infecte tous les nœuds de cette data set

Conclusion

D'après les cinq exemples qu'on a traite nous voyons que le k-means avec score ou le k-mean travaille beaucoup mieux sur les petits data sets qui contient un nombre des nœuds pas très élevé, et pour le grands réseau centroïde nous pensons que la méthode topsis est favorable pour identifier l'influence.

Conclusion générale et perspectives

Pendant la période de la réalisation de ce projet, on a eu la chance d'appliquer notre savoir et notre connaissance acquise Durant nos études à la sainte de la faculté des sciences SAMLILIA, et acquérir de nouveaux outils de développement.

Notre projet a comme objectif de définir l'influence dans des réseaux complexe. Nous avons analysé différente réseau pour savoir l'efficacité des méthodes implémenter pendant la réalisation de ce travail.

La méthode TOPSIS a donné des résultats satisfait dans les grands réseaux Parlant sur les petits réseaux, l'algorithme de k-means est plus adapte pour ce genre ; les centroïdes qui résulte avec-il, peut propage mieux l'information à l'intérieur de clusters qui constituent le réseau.

A la fin de ce travail nous avons développé une fonction à l'intérieur de l'algorithme k_means pour calculer la distance entre deux nœuds, par l'utilisation des scores d'un nœud, elle a donné des bons résultats dans les petits réseaux.

La réalisation de chaque projet rencontre des difficultés qui nous poussent a donné le meilleur, et d'apprendre des nouvelles connaissances, parmi ces difficultés, nous sommes des débutons dans le domaine d'intelligence artificielle, et de programmer avec des nouveaux langages quand n'avons pas l'habitude de l'utiliser. Malgré tout ça ce projet a été une excellente expérience pour nous.

En termes de perspectives, on pourra utiliser k-means dynamique pour trouver le nombre de k optimal s'adaptant à la taille du réseau. Comme on compte utiliser des techniques d'apprentissage par renforcement qui permettent de remplacer l'intervention humaine.

Reference

https://www.sciencedirect.com/science/article/pii/S0378437113011552

https://www.youtube.com/c/MachineLearnia

https://github.com/MachineLearnia/Python-Machine-Learning

https://www.mathweb.fr/euclide/les-graphes-en-python/

https://tel.archives-ouvertes.fr/tel-00460708

https://datascientest.com/apprentissage-non-supervise

https://aclanthology.org/2018.jeptalnrecital-recital.9.pdf

https://sites.google.com/a/uca.ma/sadgal/enseignement/enseiglpi-ia

https://tel.archives-ouvertes.fr/tel-03640442/document

https://github.com/ChekrounMohammed/Identification-of-top-k-nodes-using-TOPSIS-method-and-community-detection-in-complex-Networks

https://github.com/achrafs758/A-dynamic-weighted-TOPSIS-method-for-identifying-influential-nodes-in-complex-networks/blob/main/Rapport_Topsis_final.pdf

https://snap.stanford.edu/data/ego-Facebook.html

http://konect.cc/networks/